

**Řešení symetrického problému  
vlastních čísel se zaměřením na  
dynamiku železničních kol**

**Solution of the symmetric  
eigenvalue problem with application  
to the analysis of vibration of  
wheels**

# Zadání bakalářské práce

Student:

**Josef Raška**

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

1103R031 Výpočetní matematika

Téma:

**Řešení symetrického problému vlastních čísel se zaměřením na  
dynamiku železničních kol  
Solution of the symmetric eigenvalue problem with application to the  
analysis of vibrations of wheels**

Zásady pro vypracování:

Cílem diplomové práce je pochopit motivaci problému vlastních čísel v oblastech výzkumu souvisejících s aktivitou na Katedře aplikované matematiky VŠB (se zaměřením na analýzu výsledků experimentálního výzkumu vlastních kmitů železničních kol), prostudovat základní techniky výpočtu vlastních čísel symetrických pozitivně definitních matic, zejména jejich předpodmíněných verzí, naprogramovat je a porovnat je na jednoduchých testovacích příkladech.

Postup práce

1. Prostudovat doporučenou literaturu.
2. Vyhledat aplikace.
3. Naprogramovat vybrané metody v MATLABU.
4. Provést numerické experimenty.
5. Zpracovat výsledky.

The student should follow the following scheme:

1. Read the references.
2. Find applications related to the research of the department.
3. Code chosen algorithms in MATLAB.
4. Carry out numerical experiments.
5. Process the results

Seznam doporučené odborné literatury:

1. Y. Saad, Numerical Methods for Large eigenvalue Solvers, SIAM, Philadelphia 2010.
2. A. Knyazev, Preconditioned eigensolvers – an oxymoron?
3. Internetové zdroje

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **prof. RNDr. Zdeněk Dostál, DSc.**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013

*Bouchala*

doc. RNDr. Jiří Bouchala, Ph.D.  
vedoucí katedry



*Snášel*

prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7. května 2013

*Joel Rosta* .....

Rád bych na tomto místě poděkoval vedoucímu mé práce prof. RNDr. Zdeňku Dostálovi, DSc. za jeho trpělivost a ochotu během vytváření mé bakalářské práce.  
Dále bych chtěl poděkovat mé přítelkyni Kláře a rodině za lásku a podporu, díky které se mi podařilo moji práci dokončit.

## Abstrakt

Práce obsahuje úvod k problematice vlastních čísel, pojednává o možnostech jejich využití a ukazuje některé metody řešení tohoto problému. Jsou zde uvedeny tři příklady aplikace vlastních čísel na reálnou úlohu, které ukazují, že se nejedná pouze o teoretický problém a je třeba mu věnovat pozornost. Dále se práce zabývá představením několika metod pro výpočet vlastních čísel, z nichž jsou některé implementovány v Matlabu a následně otestovány. Výsledky jsou analyzovány a na jejich základě je předchozí implementace vylepšena. V závěru práce je uveden nástin nové větve metod, a to předpodmíněné řešiče vlastních čísel, které budou předmětem dalšího výzkumu.

**Klíčová slova:** vlastní číslo, vlastní vektor, vlastní dvojice, aplikace vlastních čísel, mocninná metoda, Lanczosova metoda, předpodmínění

## Abstract

This thesis contains an introduction to the eigenvalue problem. It discusses the possibilities of its use and shows some methods on how this problem can be solved. There are three examples where the eigenvalues are applied to real situations, which show that this is not just a theoretical problem and attention must be paid to it. The work deals with the performance of several methods for calculating eigenvalues, some of which implemented in Matlab and then tested. The results are analyzed and based on them, the previous implementation is being improved. At the end of the work, an outline of a new branch of methods is given, with preconditioned eigensolvers, which will be the subject of further research.

**Keywords:** eigenvalue, eigenvector, eigenpair, applications of eigenvalues, power method, Lanczos method, preconditioning

## Seznam použitých zkratk a symbolů

$A^{-1}$	– Inverzní matice k matici $A$
$A^T$	– Transponovaná matice $A$
$I$	– Jednotková matice
$\det(A)$	– Determinant matice $A$
$a_{ij}$	– Prvek matice $A$ na $i$ -tém řádku a $j$ -tém sloupci
$\ v\ $	– Eukleidovská norma vektoru $v$
$(a, b)$	– Skalární součin vektorů $a$ a $b$
$k$	– Tuhost pružiny v $\text{N}\cdot\text{m}^{-1}$
$m$	– Hmotnost v kg
$o$	– Nulový vektor
$\omega$	– Úhlová frekvence v $\text{s}^{-1}$
$\sigma(A)$	– Množina všech vlastních čísel matice $A$ - spektrum

## Obsah

<b>1</b>	<b>Úvod</b>	<b>5</b>
<b>2</b>	<b>Základní definice vlastních čísel</b>	<b>6</b>
2.1	Vlastnosti spektra . . . . .	6
2.2	Symetrické matice . . . . .	6
<b>3</b>	<b>Motivace a aplikace vlastních čísel na reálné problémy</b>	<b>8</b>
3.1	Vibrace . . . . .	8
3.2	Průhyb prutu . . . . .	10
3.3	Markovův řetězec . . . . .	17
<b>4</b>	<b>Metody výpočtu</b>	<b>21</b>
4.1	Charakteristický polynom . . . . .	21
4.2	Geršgorinova věta . . . . .	23
4.3	Podobnost matic . . . . .	24
4.4	Mocninná metoda . . . . .	25
4.5	Lanczosova metoda . . . . .	29
4.6	Numerické experimenty . . . . .	32
<b>5</b>	<b>Další výzkum - Předpodmíněné řešiče vlastních čísel</b>	<b>38</b>
5.1	Předpodmínění soustav lineárních rovnic . . . . .	38
5.2	Předpodmínění u vlastních čísel . . . . .	38
<b>6</b>	<b>Závěr</b>	<b>39</b>
<b>7</b>	<b>Reference</b>	<b>40</b>
	<b>Přílohy</b>	<b>40</b>
<b>A</b>	<b>Zdrojové kódy v Matlabu</b>	<b>41</b>
A.1	Základní implementace vybraných metod . . . . .	41
A.2	Získání více dominantních vlastních čísel . . . . .	43



## Seznam obrázků

1	Systém pružin . . . . .	8
2	Nosník . . . . .	10
3	Řešení příkladu 3.1 . . . . .	16
4	Jednoduchá ilustrace Markovova řetězce . . . . .	17
5	Pravděpodobnosti přechodů stavů . . . . .	18
6	Lokalizace spektra . . . . .	24
7	Chyby výpočtu algoritmu 4.6 . . . . .	35
8	Porovnání časů algoritmu 4.8 a 4.6 . . . . .	37

## Seznam tabulek

1	Výsledky testu algoritmu 4.6 pro matici $225 \times 225$ . . . . .	34
2	Výsledky testu algoritmu 4.7 pro matici $225 \times 225$ . . . . .	34
3	Výsledky testu algoritmu 4.6 pro matici $900 \times 900$ . . . . .	34
4	Výsledky testu algoritmu 4.7 pro matici $900 \times 900$ . . . . .	34
5	Výsledky testu algoritmu 4.6 pro matici $2500 \times 2500$ . . . . .	34
6	Výsledky testu algoritmu 4.7 pro matici $2500 \times 2500$ . . . . .	35
7	Výsledky testu algoritmu 4.8 pro matici $2500 \times 2500$ . . . . .	36

## Seznam výpisů zdrojového kódu

1	Algoritmus mocninné metody . . . . .	41
2	Algoritmus Iterace Rayleighova kvocientu s aproximací mocninnou meto- dou . . . . .	42
3	Lanczosova metoda . . . . .	43
4	Získání vlastního čísla z matice a vektoru . . . . .	43
5	Výpočet projektoru . . . . .	44
6	Získání $n$ dominantních vlastních čísel a vektorů mocninnou metodou . .	44
7	Získání $n$ dominantních vlastních čísel a vektorů iterací Rayleighova kvo- cientu . . . . .	45
8	Získání $n$ největších a nejmenších vlastních čísel a vektorů mocninnou me- todou a posunutím . . . . .	46
9	Získání $n$ největších a nejmenších vlastních čísel a vektorů Iterací Raylei- ghova kvocientu a posunutím . . . . .	47
10	Získání $n$ největších vlastních čísel mocninnou metodou pro řídké matice	48

## 1 Úvod

Téma výpočtu vlastních čísel se v současnosti vyskytuje napříč všemi, nejen čistě technickými, obory. Problém vlastních čísel se kromě mechaniky objevuje v elektrotechnice, makroekonomice, informatice nebo chemii. Takovéto množství oborů přináší spoustu různých technických úloh, vedoucích na řešení problému vlastních čísel a se zvětšujícími se rozsahy problémů roste pochopitelně náročnost řešení. Spektrální teorie je proto již několik desetiletí rychle rozvíjena a s využitím moderní techniky lze dnes řešit problémy obrovských rozměrů. Jak už to však bývá, výpočetní síla počítačů nebude nikdy zcela dostačující a jsou vyvíjeny stále lepší a efektivnější metody, jak problém vlastních čísel řešit.

Na následujících stránkách si nejprve demonstrujeme na příkladech, kde se můžou vlastní čísla vyskytnout a jak je lze využít v praxi. Poté vedeme několik metod pro výpočet vlastních čísel spolu s jejich implementací v Matlabu, z nichž některé otestujeme a porovnáme. Na závěr nastíníme předmět dalšího výzkumu, a to předpokládané řešiče, které jsou v této problematice nejnovějším nástrojem.

## 2 Základní definice vlastních čísel

Při studiu mnoha inženýrských problémů se můžeme často setkat s problémem vlastních čísel a vektorů. Ty jsou definovány následovně:

Nechť  $U$  je vektorový prostor. Mějme lineární transformaci  $A : U \rightarrow U$ , nenulový vektor  $v \in U$  a číslo  $\lambda \in \mathbb{C}$ , a platí

$$Av = \lambda v, \quad (2.1)$$

pak číslo  $\lambda$  nazýváme vlastním číslem a vektor  $v$  jemu příslušný vlastní vektor zobrazení  $A$ . Dvojici  $(\lambda, v)$  pak nazýváme vlastní dvojicí zobrazení  $A$ . V této práci se budeme zabývat zejména vlastními čísly matic a  $A$  budeme používat pro označení komplexní čtvercové matice.

Rovnici (2.1) můžeme přepsat do tvaru

$$Av - \lambda v = (A - \lambda I)v = 0. \quad (2.2)$$

Získáváme tak soustavu rovnic, jejímž řešením je vlastní vektor matice  $A$ . Jelikož vektor  $v$  nesmí být podle definice nulový, má daná soustava netriviální, a tedy nenulové řešení pouze tehdy, pokud je matice  $A - \lambda I$  singulární a tedy platí

$$\det(A - \lambda I) = 0. \quad (2.3)$$

Množinu všech vlastních čísel  $\sigma(A)$  nazýváme spektrum matice  $A$ .

Vlastní vektor  $v$  nazýváme také pravý vlastní vektor. Levý vlastní vektor definujeme rovností

$$v^T A = \lambda v^T. \quad (2.4)$$

### 2.1 Vlastnosti spektra

Spektrum má spoustu zajímavých vlastností, které nám často mohou říci mnoho o systému, reprezentovaném maticí. Samotné spektrum matice většinou není vidět, je však známo, že

$$\det(A) = \prod_{i=1}^n \lambda_i. \quad (2.5)$$

Někdy také můžeme využít stopu matice, což je součet prvků matice na hlavní diagonále a pro níž platí

$$\sum_{i=1}^n a_{ii} = \sum_{i=1}^n \lambda_i. \quad (2.6)$$

### 2.2 Symetrické matice

V této práci se budeme soustředit zejména na symetrické matice, neboť se vyskytují při vyšetřování reálných problémů velmi často a pro řešení problému vlastních čísel lze tyto vlastnosti využít.

**Věta 2.1** *Jestliže je  $A$  symetrická, reálná matice, pak platí  $\forall \lambda \in \sigma(A) : \lambda \in \mathbb{R}$ .*

**Věta 2.2** *Jestliže je  $A$  symetrická, reálná matice, pak jsou její vlastní vektory, příslušné různým vlastním číslům ortogonální.*

Důkazy těchto vět lze nalézt například v [5].

Pro tyto symetrické matice lze pak nalézt spektrální rozklad

$$A = QDQ^T, \tag{2.7}$$

kde  $D$  je matice s vlastními čísly na diagonále a  $Q$  je matice, mající ve sloupcích příslušné ortonormální vlastní vektory.

### 3 Motivace a aplikace vlastních čísel na reálné problémy

Vlastní čísla mají velmi široké uplatnění v technické praxi a mnoho odborných a technických článků se tímto tématem zabývá. V akademických materiálech se však již vyskytuje těchto ukázek méně. Z tohoto důvodu uvedeme několik reálných příkladů využití vlastních čísel.

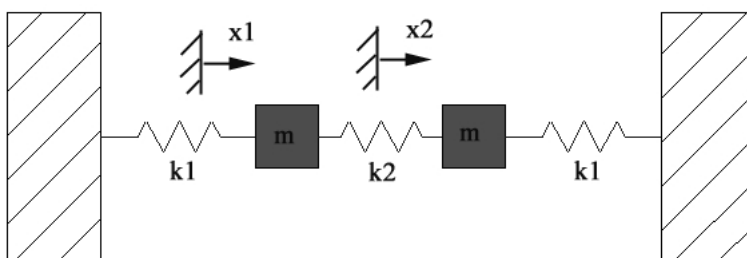
#### 3.1 Vibrace<sup>1</sup>

V mechanice často využívaná aplikace vlastních čísel. Vlastní čísla zde ukazují vlastní frekvenci konstrukce reprezentované maticí. Vlastní frekvenci se nazývá rezonanční frekvence soustavy a v praxi může buzení systému touto frekvencí způsobit rozkmitání i velkých konstrukcí a vést k jejich poškození. Typickým příkladem tohoto problému může být slavný Tacoma Bridge.

(<http://www.youtube.com/watch?v=j-zczJXSxw>)

##### 3.1.1 Vlastní frekvence

Mějme systém tří pružin a dvou hmotných objektů zobrazený na obrázku 1, pohybujících se pouze v horizontálním směru.



Obrázek 1: Systém pružin

Zdroj: <http://lpsa.swarthmore.edu/MtrxVibe/EigApp/EigVib.html>

Rovnice rovnováhy vypadají takto

$$mx_1'' + k_1x_1 - k_2(x_2 - x_1) = 0, \quad (3.1)$$

$$mx_2'' + k_1x_2 + k_2(x_2 - x_1) = 0. \quad (3.2)$$

Budeme se snažit najít řešení při buzení soustavy harmonickým kmitáním s parametry  $\omega$  a  $\varphi$

$$x_i = c_i \cos(\omega t + \varphi), \quad (3.3)$$

<sup>1</sup>Inspirace v [9]

které dále derivujeme

$$x_i' = -\omega c_i \sin(\omega t + \varphi), \quad (3.4)$$

$$x_i'' = -\omega^2 c_i \cos(\omega t + \varphi) = \underline{-\omega^2 x_i}. \quad (3.5)$$

Po dosazení do (3.1) a (3.2) upravíme rovnice

$$-\omega^2 x_1 m + k_1 x_1 - k_2(x_2 - x_1) = 0 \quad (3.6)$$

$$-\omega^2 x_1 m + k_1 x_1 + k_2 x_1 - k_2 x_2 = 0 \quad / \cdot \frac{1}{m} \quad (3.7)$$

$$-\omega^2 x_1 + \frac{k_1 x_1 + k_2 x_1}{m} - \frac{k_2 x_2}{m} = 0 \quad (3.8)$$

$$x_1 \left( -\omega^2 + \frac{k_1 + k_2}{m} \right) + x_2 \left( -\frac{k_2}{m} \right) = 0 \quad (3.9)$$

$$-\omega^2 x_2 m + k_1 x_2 + k_2(x_2 - x_1) = 0 \quad (3.10)$$

$$-\omega^2 x_2 m + k_1 x_2 + k_2 x_2 - k_2 x_1 = 0 \quad / \cdot \frac{1}{m} \quad (3.11)$$

$$-\frac{k_2 x_1}{m} - \omega^2 x_2 + \frac{k_1 x_2 + k_2 x_2}{m} = 0 \quad (3.12)$$

$$x_1 \left( -\frac{k_2}{m} \right) + x_2 \left( -\omega^2 + \frac{k_1 + k_2}{m} \right) = 0 \quad (3.13)$$

a můžeme soustavu napsat maticově

$$\begin{pmatrix} -\omega^2 + \frac{k_1 + k_2}{m} & -\frac{k_2}{m} \\ -\frac{k_2}{m} & -\omega^2 + \frac{k_1 + k_2}{m} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Pokud bude první matice, označme ji  $M$ , regulární, existuje vždy pouze jedno, nulové řešení a soustava buzená tímto kmitáním se utlumí. Pokud bude však singulární, není řešení jednoznačně určeno a existují i nenulová řešení.

Nyní si je třeba všimnout, že první matici lze napsat jako

$$M = \begin{pmatrix} \frac{k_1 + k_2}{m} & -\frac{k_2}{m} \\ -\frac{k_2}{m} & -\frac{k_1 + k_2}{m} \end{pmatrix} - \omega^2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Jak známo, matice  $M$  je singulární, pokud  $\det(M) = 0$ , tedy pokud označíme první matici jako  $A$ , pak  $\det(A - \omega^2 I) = 0$ , což přesně odpovídá definici (2.3) pro vlastní čísla a můžeme psát

$$\underline{\omega^2 = \lambda}. \quad (3.14)$$

Zde vidíme, že vlastní čísla  $\lambda$  musí být z důvodu druhé mocniny větší nebo rovna nule. Není to však žádné překvapení, neboť frekvence pochopitelně nemůže být záporná. V



našem případě budou tyto vlastní frekvence dvě a můžeme je snadno vypočítat řešením determinantu matice.

$$\det(M) = 0 \quad (3.15)$$

$$\left(-\omega^2 + \frac{k_1 + k_2}{m}\right)^2 - \frac{k_2^2}{m^2} = 0 \quad (3.16)$$

$$-\omega^2 + \frac{k_1 + k_2}{m} = \pm \frac{k_2}{m} \quad (3.17)$$

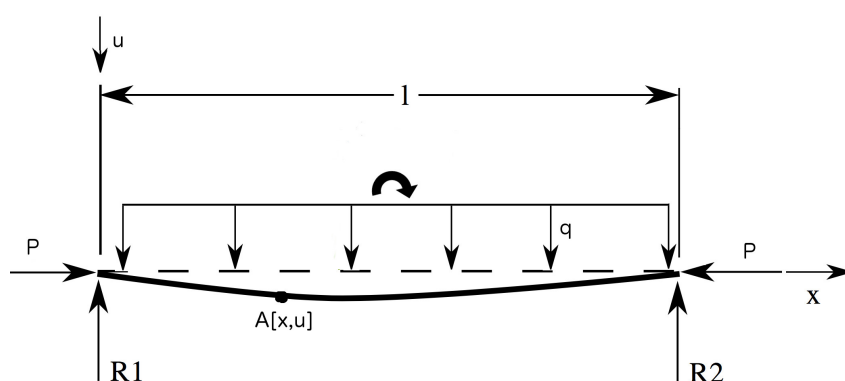
$$\omega_1^2 = \frac{k_1}{m} \rightarrow \omega_1 = \sqrt{\frac{k_1}{m}} \quad \omega_2^2 = \frac{k_1 + 2k_2}{m} \rightarrow \omega_2 = \sqrt{\frac{k_1 + 2k_2}{m}} \quad (3.18)$$

**Závěr:** Jelikož není řešení v tomto případě jednoznačně určeno, může se soustava při těchto frekvencích neočekávaně rozkmitat, což může v praxi vést k silnému namáhání konstrukce nebo dokonce k jejímu zničení.

## 3.2 Průhyb prutu <sup>2</sup>

### 3.2.1 Formulace problému

Velice hezkým příkladem aplikace vlastních čísel je hledání průhybu homogenního, vodorovného prutu o dané délce  $l$ . Prut je volně uložen na nosnících, konstantního průřezu, je rovnoměrně zatížen silou  $q$  a je namáhán na vzpěr tlakovou silou  $P$ . Nákres prutu, příslušné síly a orientace zvoleného souřadnicového systému  $x, u$ , lze vidět na obrázku 2.



Obrázek 2: Nosník

Zdroj: Upravený obrázek z <http://openclipart.org/detail/100549>

<sup>2</sup>Inspirace v [2]

Pro obecný průhyb prutu v bodě  $A[x, u]$  platí rovnice

$$u'' = -\frac{M}{EI}, \quad (3.19)$$

kde  $E$  je modul pružnosti v tahu,  $I$  je moment setrvačnosti průřezu vzhledem k ohybové ose a  $M$  je velikost momentu všech sil, působící na daný bod  $A$  zleva. Směr momentu je nakreslen na obrázku. Moment lze vyjádřit jako součet momentu reakce  $R_1$  s ramenem  $x$ , síly  $P$  s ramenem  $u$  a zatížení po délce intervalu  $\langle 0; x \rangle$ . Síla zatížení se zde rovná síle velikosti  $qx$  působící v těžišti intervalu a délka ramena je tedy  $\frac{x}{2}$ . Tato síla působí proti směru momentu, naznačeném na obrázku a je proto třeba ji od výsledného momentu odečítat.

$$M = \frac{ql}{2}x + Pu - qx\frac{x}{2} \quad (3.20)$$

Výsledná diferenciální rovnice má tvar

$$u'' = \frac{-Pu}{EI} + \frac{q}{2EI}x^2 - \frac{ql}{2EI}, \quad (3.21)$$

s okrajovými podmínkami

$$u(0) = 0, \quad u(l) = 0. \quad (3.22)$$

### 3.2.2 Analytické řešení

Pomocí substitucí

$$a^2 = \frac{-Pu}{EI}, \quad b = \frac{q}{2EI}, \quad c = -\frac{ql}{2EI}, \quad (3.23)$$

můžeme rovnici (3.21) vyjádřit ve tvaru

$$u'' + a^2u = bx^2 - cx. \quad (3.24)$$

Obecným řešením přidružené homogenní rovnice je

$$u_h = C_1 \cos(ax) + C_2 \sin(ax), \quad C_1, C_2 \in \mathbb{R}. \quad (3.25)$$

Partikulární řešení získáme metodou speciální pravé strany, kterou hledáme ve tvaru

$$u_p = Ax^2 + Bx + C, \quad u_p'' = 2A. \quad (3.26)$$

Dosazením do (3.24) získáváme následující rovnici a porovnáním koeficientů u stejných mocnin  $x$  získáme partikulární řešení.

$$a^2Ax^2 = bx^2 + cx \quad (3.27)$$

$$a^2 Ax^2 = bx^2 \rightarrow A = \frac{b}{a^2} \quad (3.28)$$

$$a^2 Bx = bx^2 \rightarrow B = \frac{c}{a^2} \quad (3.29)$$

$$2A + a^2 C = \frac{2b}{a^2} + a^2 C = 0 \rightarrow C = -\frac{2b}{a^4} \quad (3.30)$$

$$u_p = \frac{b}{a^2} x^2 + \frac{c}{a^2} x - \frac{2b}{a^4} \quad (3.31)$$

Sečtením homogenního a partikulárního řešení získáváme obecné řešení (3.24).

$$u = \frac{b}{a^2} x^2 + \frac{c}{a^2} x - \frac{2b}{a^4} + C_1 \cos(ax) + C_2 \sin(ax), \quad C_1, C_2 \in R \quad (3.32)$$

Dále aplikujeme počáteční podmínky (3.22) pro získání konstant  $C_1$  a  $C_2$ . Pro  $x = 0$  platí

$$0 = -\frac{2b}{a^4} + C_1 \rightarrow C_1 = \frac{2b}{a^4}. \quad (3.33)$$

Pro  $x = l$  získáváme rovnici

$$0 = \frac{b}{a^2} l^2 + \frac{c}{a^2} l - \frac{2b}{a^4} + \frac{2b}{a^4} \cos(al) + C_2 \sin(al). \quad (3.34)$$

Jak lze vidět v (3.21),  $c = -bl$  a platí

$$\frac{b}{a^2} l^2 + \frac{c}{a^2} l = 0 \quad (3.35)$$

$$0 = \frac{2b}{a^4} l (\cos(al) - 1) + C_2 \sin(al) \quad (3.36)$$

$$C_2 = \frac{2b}{a^4} \frac{(1 - \cos(al))}{\sin(al)}, \quad \sin(al) \neq 0. \quad (3.37)$$

Řešení (3.24) s počátečními podmínkami (3.22), a tedy průhyb prutu vypadá takto

$$u = \frac{b}{a^2} x^2 + \frac{c}{a^2} x - \frac{2b}{a^4} + \frac{2b}{a^4} \cos(ax) + \frac{2b}{a^4} \frac{(1 - \cos(al))}{\sin(al)} \sin(ax). \quad (3.38)$$

### 3.2.3 Kritická síla

Při výpočtu druhé konstanty je nutné zavést podmínku  $\sin(al) \neq 0$ , z důvodu výskytu tohoto výrazu ve jmenovateli. Pro všechny případy, kdy  $al$  je násobek  $\pi$ , se stává úloha s okrajovými podmínkami neřešitelnou. Je třeba si však uvědomit, co to znamená v praxi. Pokud se podíváme na rovnici (3.38), vidíme, že všechny členy rovnice kromě posledního budou vždy nabývat konečných hodnot. Pokud se však bude jmenovatel posledního výrazu blížit k nule, průhyb prutu začne růst nade všechny meze. Nastane přerušení a prut praskne. Z uvedené podmínky můžeme získat tak zvanou kritickou sílu.

$$\sin(al) = 0 \quad (3.39)$$

$$al = \pi \quad (3.40)$$

$$a^2 l^2 = \pi^2 \quad (3.41)$$

$$\frac{P}{EI} l^2 = \pi^2 \quad (3.42)$$

$$P_{kr} = \frac{\pi^2 EI}{l^2} \quad (3.43)$$

$$(3.44)$$

Je důležité si všimnout, že kritická síla nezávisí na vertikálním zatížení, ale pouze na řešení přidružené homogenní rovnice (konstantu  $C_2$  nelze vypočítat při  $\sin(al) = 0$ ). Kritická síla se tudíž vyskytuje i v následujícím případě s nulovým zatížením.

$$u'' + a^2 u = 0, \quad u(0) = 0, \quad u(l) = 0 \quad (3.45)$$

Obecné řešení vypadá stejně jako v (3.25) takto

$$u_h = C_1 \cos(ax) + C_2 \sin(ax), \quad C_1, C_2 \in R. \quad (3.46)$$

Z první podmínky je vidět, že  $C_1 = 0$ . Mnohem zajímavější je však výpočet druhé konstanty

$$C_2 \sin(al) = 0. \quad (3.47)$$

Zde platí, že pro  $al \in (0, \pi)$ ,  $C_2 = 0$ . Úloha (3.45) tak získává jediné řešení

$$u = 0. \quad (3.48)$$

Pokud však  $al = \pi$ , je hodnota  $C_2$  neurčitá a řešení není dáno jednoznačně. Plyne z toho následující poznatek. V případě nenulového vertikálního zatížení je průhyb prutu konečný, při nulovém zatížení je neurčitý. Jako typický příklad může sloužit nosný sloup konstrukce. Sloup je zde ve vertikální poloze a je namáhán tíhou konstrukce, kterou nese. Zatížení ze strany je tedy nulové. Pokud však síla působící na sloup ve vertikálním směru dosáhne  $P_{kr}$ , není průhyb udán jednoznačně. Průhyb v horizontálním směru může zůstat nulový, stačí však jakékoliv horizontální zatížení a sloup se zřítí. Zhroucení konstrukce může způsobit v tomto případě i závan větru.

### 3.2.4 Spojení a problémem vlastních čísel

Jak můžeme vidět v předchozím příkladu, problém s řešením nastává pouze v případě, pokud má příslušná homogenní rovnice jiné než nulové řešení. Mějme problém

$$u'' + \lambda u = 0, \quad u(0) = 0, \quad u(l) = 0, \quad (3.49)$$

kde  $\lambda$  je reálný parametr. Řešením tohoto problému je spojitá funkce v intervalu, která vyhovuje okrajovým podmínkám a má z důvodu prvního členu v intervalu spojitě první dvě derivace. Problém má samozřejmě triviální řešení. Existují však hodnoty  $\lambda$ , pro které existuje také netriviální řešení. Tato čísla se nazývají vlastní čísla problému (3.49) a funkci, řešící problém s tímto parametrem nazýváme vlastní funkcí.

### 3.2.5 Výpočet vlastních čísel a řešení obyčejných diferenciálních rovnic metodou sítí

Výše popsany problém lze řešit numericky, metodou konečných diferencí neboli metodou sítí. Ta funguje na principu rozdělení daného intervalu na  $n$  stejně velkých podintervalů délky  $h$  a aproximací derivací pomocí funkčních hodnot v okolních bodech. Odvození metody sítí, které lze nalézt v [4], pomocí Taylorových polynomů druhého stupně vypadá následovně.

$$u(x+h) = u(x) + \frac{h}{1!}u'(x) + \frac{h^2}{2!}u''(x) + C_1(h^3) \quad (3.50)$$

$$u(x-h) = u(x) - \frac{h}{1!}u'(x) + \frac{h^2}{2!}u''(x) + C_2(h^3) \quad (3.51)$$

Po sečtení těchto dvou rovnic a zanedbání řádů vyšších než 2 dostáváme

$$u(x+h) + u(x-h) \approx 2u(x) + h^2u'' \quad (3.52)$$

$$u'' \approx \frac{u(x+h) + u(x-h) - 2u(x)}{h^2}. \quad (3.53)$$

**Příklad 3.1** Mějme problém

$$u'' - (1 - \cos^2(x))u = -2 \quad (3.54)$$

s okrajovými podmínkami

$$u(0) = 0, \quad u(\pi) = 0 \quad (3.55)$$

a počtem podintervalů  $n = 6$ ,  $h = \frac{\pi}{6}$ . Položme  $g(x) = -(1 - \cos^2 x)$  a  $f(x) = -2$ .

**Řešení:** Obecně lze pomocí metody sítí průhyb v bodě  $x_i$  vyjádřit takto

$$f(x_i) = \frac{u(x_{i-1}) - 2u(x_i) + u(x_{i+1}))}{h^2} + g(x_i)u(x_i), \quad (3.56)$$

$$f(x_i)h^2 = u(x_{i-1}) - 2u(x_i) + u(x_{i+1}) + g(x_i)u(x_i)h^2. \quad (3.57)$$

Přičemž podle okrajových podmínek  $u(x_0) = 0$  a  $u(x_n) = 0$ .

Vznikající matici můžeme sestavit jako součet matic  $A = L + G$ .

$$L = \begin{pmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \ddots & \vdots \\ 0 & 1 & -2 & 1 & \ddots \\ & \ddots & 1 & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & -2 & 1 \\ 0 & \cdots & & 0 & 1 & -2 \end{pmatrix} \quad G = h^2 \begin{pmatrix} g(x_1) & 0 & \cdots & 0 \\ 0 & g(x_2) & 0 & & \\ & 0 & g(x_3) & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots \\ 0 & & \cdots & g(x_{n-2}) & 0 & g(x_{n-1}) \end{pmatrix}$$

Pravá strana  $b$  a vektor neznámých  $u_1$  až  $u_{n-1}$  vypadají následovně

$$b = \begin{pmatrix} f(x_1)h^2 - u_0 \\ f(x_2)h^2 \\ f(x_3)h^2 \\ \vdots \\ f(x_{n-2})h^2 \\ f(x_{n-1})h^2 - u_n \end{pmatrix}, \quad u = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{n-2} \\ u_{n-1} \end{pmatrix}.$$

Po dosazení a využití vztahu  $1 - \cos^2 x = \sin^2 x$  vypadá soustava takto

$$A = \begin{pmatrix} -2 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -2 \end{pmatrix} - \frac{\pi^2}{36} \begin{pmatrix} \sin^2 \frac{\pi}{6} & 0 & 0 & 0 & 0 \\ 0 & \sin^2 \frac{\pi}{3} & 0 & 0 & 0 \\ 0 & 0 & \sin^2 \frac{\pi}{2} & 0 & 0 \\ 0 & 0 & 0 & \sin^2 \frac{2\pi}{3} & 0 \\ 0 & 0 & 0 & 0 & \sin^2 \frac{5\pi}{6} \end{pmatrix}, \quad b = -\frac{\pi^2}{18} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix},$$

kterou vyřešíme a získáváme výsledné řešení  $u$ .

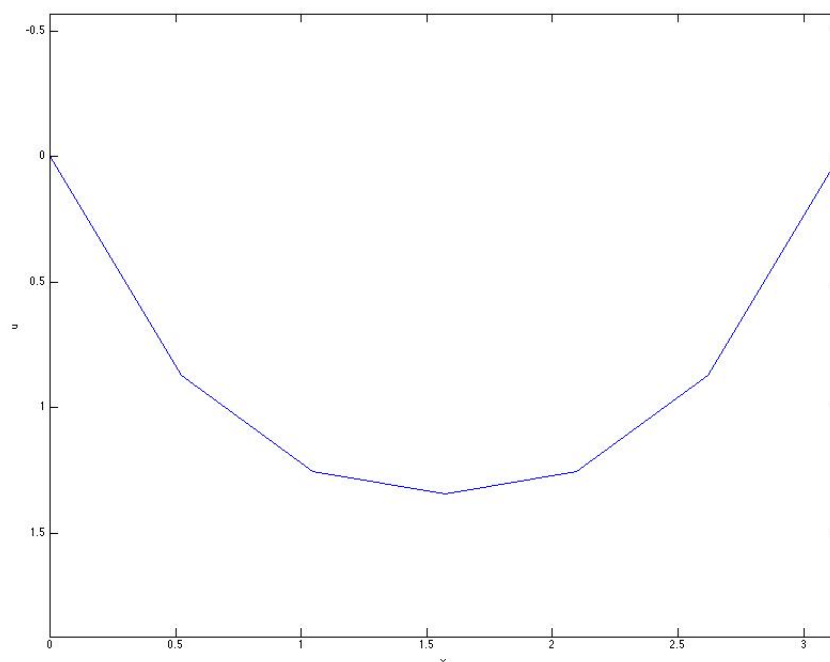
$$u = (0,8700 \quad 1,2513 \quad 1,3415 \quad 1,2513 \quad 0,8700)^T$$

**Příklad 3.2** Vypočítejme prvních 5 vlastních čísel problému

$$u'' - \lambda(1 - \cos^2 x)u = -2 \quad (3.58)$$

s okrajovými podmínkami

$$u(0) = 0, \quad u(\pi) = 0. \quad (3.59)$$



Obrázek 3: Řešení příkladu 3.1

**Řešení:** Hledáme parametry  $\lambda$  rovnice

$$u'' + \lambda g(x)u = 0, \quad (3.60)$$

kde  $g(x) = \cos^2 x - 1$  a pro které má tato rovnice jiné, než triviální řešení. To znamená, že sestavená matice  $A$  pomocí metody sítí je singulární a tedy  $\det(A) = 0$ . Matice  $L$  a  $G$  vypadají stejně, jako v předchozím případě, s tím rozdílem, že matice  $G$  se násobí parametrem  $\lambda$ :  $A = L + \lambda G$ . Nyní můžeme pomocí pravidel pro počítání s determinanty (např. v [5]) odvodit postup pro výpočet vlastních čísel.

$$\det(L + \lambda G) = 0 \quad / \cdot \det(G^{-\frac{1}{2}}) \text{ zprava} \quad (3.61)$$

$$\det(LG^{-\frac{1}{2}} + \lambda G^{\frac{1}{2}}) = 0 \quad / \cdot \det(G^{-\frac{1}{2}}) \text{ zleva} \quad (3.62)$$

$$\det(G^{-\frac{1}{2}}LG^{-\frac{1}{2}} + \lambda I) = 0 \quad / \cdot (-1) \quad (3.63)$$

$$\det(-G^{-\frac{1}{2}}LG^{-\frac{1}{2}} - \lambda I) = 0 \quad (3.64)$$

Podle definice vlastních čísel je tedy z rovnice (3.64) vidět, že vlastní čísla problému (3.60) jsou vlastní čísla matice  $G^{-\frac{1}{2}}LG^{-\frac{1}{2}}$ . V úpravách byla používána matice  $G^{-\frac{1}{2}}$  z důvodu zachování symetričnosti výsledné matice.

Sestavená matice pro problém (3.60) vypadá následovně

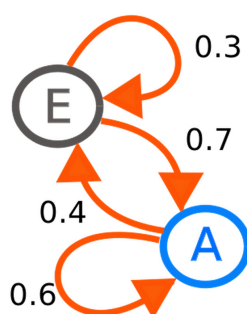
$$-G^{-\frac{1}{2}}LG^{-\frac{1}{2}} = \begin{pmatrix} -29.1805 & 8.4237 & 0 & 0 & 0 \\ 8.4237 & -9.7268 & 4.2118 & 0 & 0 \\ 0 & 4.2118 & -7.2951 & 4.2118 & 0 \\ 0 & 0 & 4.2118 & -9.7268 & 8.4237 \\ 0 & 0 & 0 & 8.4237 & -29.1805 \end{pmatrix}$$

a vlastní čísla této matice a tedy vlastní čísla problému (3.60) s funkcí  $g(x) = \cos^2 x - 1$  jsou

$$e = (-1.2825 \quad -6.5863 \quad -12.4185 \quad -32.3211 \quad -32.5014).$$

### 3.3 Markovův řetězec

Markovův řetězec tvoří posloupnost stavů náhodného procesu, kde to, v jakém stavu se objekt bude nacházet v následujícím kroku, ovlivňuje pouze stav současný. Markovův řetězec nachází uplatnění ve fyzice, chemii, ekonomii, statistice a mnoha dalších oborech.



Obrázek 4: Jednoduchá ilustrace Markovova řetězce

Zdroj: [http://cs.wikipedia.org/wiki/Soubor:Markovkate\\_01.svg](http://cs.wikipedia.org/wiki/Soubor:Markovkate_01.svg)

Jak lze vidět na obrázku 4, pokud je objekt ve stavu E, přejde do stavu A s pravděpodobností 0.7 a zůstane ve stejném stavu s pravděpodobností 0.3. Podobně objekt ve stavu A přejde do stavu E s pravděpodobností 0.4 a s pravděpodobností 0.6 se jeho stav nezmění. Jak vidíme, tak následující stav opravdu závisí pouze na konkrétním stavu.

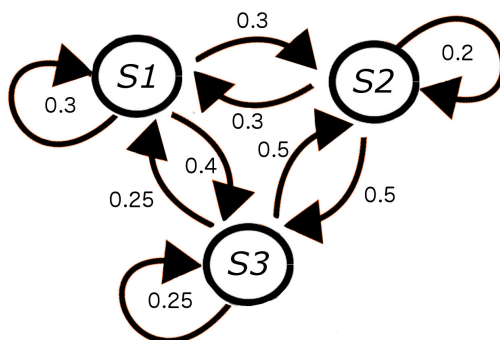
#### 3.3.1 Definice

Markovův řetězec je definován množinou  $n$  stavů  $S = \{s_1, s_2, \dots, s_n\}$  a maticí přechodů  $P$ , kde  $p_{ij}$  je pravděpodobnost přechodu ze stavu  $s_i$  do  $s_j$ .



### 3.3.2 Příklad

Mějme přepravní firmu vozící zboží ze stanovišť  $s_1, s_2$  a  $s_3$ . Poté, co je zboží doručeno se vůz firmy vrátí na nejbližší ze tří stanovišť. Podle statistik můžeme říci, že známe pravděpodobnost toho, na které stanoviště  $s_j$  a se vůz vrátí, pokud vyjede ze stanoviště  $s_i$ . Jednotlivé pravděpodobnosti přechodů jsou na obrázku 5.



Obrázek 5: Pravděpodobnosti přechodů stavů

Matice přechodů vypadá takto

$$P = \begin{matrix} & \begin{matrix} s_1 & s_2 & s_3 \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix} & \begin{pmatrix} 0.3 & 0.3 & 0.4 \\ 0.3 & 0.2 & 0.5 \\ 0.25 & 0.5 & 0.25 \end{pmatrix} \end{matrix}.$$

Nyní si uvědomme, že pravděpodobnost toho, že vůz ze stanoviště 1 se po dvou výjezdech zase vrátí, odpovídá součtu pravděpodobností těchto přechodů:  $s_1 \rightarrow s_1 \rightarrow s_1, s_1 \rightarrow s_2 \rightarrow s_1, s_1 \rightarrow s_3 \rightarrow s_1$ , což lze vyjádřit

$$p_{11}^2 = p_{11}p_{11} + p_{12}p_{21} + p_{13}p_{31}. \quad (3.65)$$

Obecně lze pravděpodobnost toho, že bude vůz ze stanoviště  $s_i$  bude po dvou výjezdech nacházet ve stavu  $s_j$  vyjádřit následovně

$$p_{ij}^2 = \sum_{k=1}^n p_{ik}p_{kj} \quad (3.66)$$

a tedy se nejedná nic jiného, než prosté násobení matic. Pro  $r$  přechodů lze psát

$$p_{ij}^r = \sum_{k=1}^n p_{ik}^{r-1}p_{kj}. \quad (3.67)$$

Pravděpodobnosti všech přechodů po  $r$  krocích získáme prostým vynásobením matice  $P$   $r$ -krát sama sebou.

$$P^2 = \begin{matrix} & \begin{matrix} s_1 & s_2 & s_3 \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix} & \begin{pmatrix} 0.2800 & 0.3500 & 0.3700 \\ 0.2750 & 0.3800 & 0.3450 \\ 0.2875 & 0.3000 & 0.4125 \end{pmatrix} \end{matrix}$$

Podobně můžeme pokračovat a získat matici pro libovolný počet přechodů.

$$P^4 = \begin{matrix} & \begin{matrix} s_1 & s_2 & s_3 \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix} & \begin{pmatrix} 0.2810 & 0.3420 & 0.3770 \\ 0.2807 & 0.3442 & 0.3752 \\ 0.2816 & 0.3384 & 0.3800 \end{pmatrix} \end{matrix}$$

$$P^8 = \begin{matrix} & \begin{matrix} s_1 & s_2 & s_3 \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix} & \begin{pmatrix} 0.2811 & 0.3414 & 0.3775 \\ 0.2811 & 0.3414 & 0.3775 \\ 0.2811 & 0.3413 & 0.3775 \end{pmatrix} \end{matrix}$$

$$P^{16} = \begin{matrix} & \begin{matrix} s_1 & s_2 & s_3 \end{matrix} \\ \begin{matrix} s_1 \\ s_2 \\ s_3 \end{matrix} & \begin{pmatrix} 0.2811 & 0.3414 & 0.3775 \\ 0.2811 & 0.3414 & 0.3775 \\ 0.2811 & 0.3414 & 0.3775 \end{pmatrix} \end{matrix}$$

Násobky matic nám rychle konvergují a po dalších násobeních se již výsledek nemění. To znamená, že pravděpodobnost toho, že bude vozidlo po  $n$  výjezdech v některém ze stanišť, nezáleží na tom, ze kterého stanoviště vyjela, a po určité době zůstanou vozidla přesně rozdělena podle daného poměru.

### 3.3.3 Vlastní čísla

Pro dostatečně velké  $n$  lze psát<sup>3</sup>

$$P^{n+1} = PP^n = P^n. \quad (3.68)$$

Pokud si matici  $P^n$  napíšeme jako sloupcové vektory  $P^n = (v_1, v_2, \dots, v_k)$ , pak pro každý z nich platí.

$$Pv_i = v_i, \quad (3.69)$$

což přesně odpovídá definici (2.1) a matice  $P$  má vlastní číslo  $\lambda = 1$  a  $v_i$  je příslušným vlastním vektorem. Lze si všimnout že všechny vektory  $v_i$  jsou pouze násobky jednotkového vlastního vektoru. Toto vlastní číslo je zároveň dominantním vlastním číslem, jak můžeme jednoduše ověřit například pomocí Geršgorinovy věty (kapitola 4.2), kde by bylo vidět, že žádné vlastní číslo nemůže být větší než 1.

Jelikož víme, že pro čtvercové matice  $A, B$ , stejné dimenze platí

$$A^T B^T = (BA)^T \quad (3.70)$$

a můžeme psát

$$P^T P^T = (PP)^T \rightarrow (P^T)^n = (P^n)^T. \quad (3.71)$$

<sup>3</sup>Důkaz tohoto tvrzení by vypadal stejně, jako důkaz konvergence mocninné metody v kapitole 4.4

Matice  $(P^T)^n$  má při dalším násobení  $P^T$  stejné vlastnosti, jako matice  $P^n$  v (3.68) a tedy také vlastní číslo  $\lambda = 1$ . Matice

$$(P^{16})^T = \begin{pmatrix} 0.2811 & 0.2811 & 0.2811 \\ 0.3414 & 0.3414 & 0.3414 \\ 0.3775 & 0.3775 & 0.3775 \end{pmatrix} \quad (3.72)$$

nám ilustruje, že matice  $(P^T)^n$  má opět ve sloupcích vlastní vektor příslušný tomuto vlastnímu číslu, tentokrát je to však už rovnou vektor pravděpodobností pro přechody.

### 3.3.4 Závěr

Pokud máme k dispozici matici přechodů nějakého procesu, můžeme získat matici přechodů po  $n$  krocích, kde  $n \rightarrow \infty$ , získáním vlastního vektoru  $v_1$  matice  $P^T$ , příslušnému vlastnímu číslu  $\lambda_1 = 1$ , pro který platí  $\|v_1\|_1 = 1$ . Vyhýbáme se tak násobení matic, které může v některých případech, zejména u matic vysokých dimenzí, konvergovat velmi pomalu.

## 4 Metody výpočtu

### 4.1 Charakteristický polynom

Charakteristickým polynomem označujeme polynom stupně  $n$ , který vznikne při výpočtu determinantu matice  $A - \lambda I$  dimenze  $n$ .

$$p_n(\lambda) = \det(A - \lambda I) \quad (4.1)$$

Položíme-li tento polynom roven nule, získáváme charakteristickou rovnici problému, kde jsou vlastní čísla  $\lambda$  kořeny této rovnice a problém vlastních čísel můžeme takto vyřešit analyticky.

$$p_n(\lambda) = 0 \quad (4.2)$$

**Příklad 4.1** *Nalezněte vlastní čísla matice*

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 2 \end{pmatrix}.$$

**Řešení:**

$$\det(A - \lambda I) = 0 \quad (4.3)$$

$$\det \begin{pmatrix} 1 - \lambda & 2 \\ 3 & 2 - \lambda \end{pmatrix} = 0 \quad (4.4)$$

$$(1 - \lambda)(2 - \lambda) - 6 = 0 \quad (4.5)$$

$$\lambda^2 - 3\lambda - 4 = 0 \quad (4.6)$$

$$(\lambda + 1)(\lambda - 4) = 0 \quad (4.7)$$

Vlastní čísla matice  $A$  jsou tedy  $\lambda_1 = -1$ ,  $\lambda_2 = 4$ .

#### 4.1.1 Základní věta algebry

Pomocí charakteristické rovnice lze tedy vždy získat vlastní čísla matice, samozřejmě za předpokladu, že ji umíme vyřešit. Jak již bylo řečeno, při výpočtu determinantu čtvercové matice dimenze  $n$  získáme charakteristický polynom řádu  $n$ . Díky základní větě algebry víme, že charakteristická rovnice má vždy alespoň jedno řešení.

**Věta 4.1** (Základní věta algebry) *Mějme polynom proměnné  $x$  a řádu  $n$  ve tvaru*

$$p_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0, \quad (4.8)$$

*kde  $a_0, \dots, a_n \in \mathbb{C}$  a  $a_n \neq 0$ . Potom existuje číslo  $a \in \mathbb{C}$  takové, že platí  $p_n(a) = 0$ .*

### 4.1.2 Hledání kořenů

Jelikož stupeň polynomu roste úměrně dimenzi matice a od stupně 5 již neznáme postup pro analytické řešení, je nutné hledání kořene polynomu řešit numericky, například metodu půlení intervalu, prostých iterací nebo Newtonovou metodou. O těchto metodách se můžeme dočíst například v [3].

Podobně nalezení samotného charakteristického polynomu může být značně netriviální a zdoluhavá úloha. K efektivnějšímu získání charakteristického polynomu lze použít Krylovovu metodu, popsanou rovněž v [3].

V praxi se však často setkáváme se speciálními typy matic, z nichž jeden je matice třídiagonální. Pro tento typ matice můžeme získat charakteristický polynom rekurzivně. Třídiagonální matice dimenze  $n$  vypadá takto:

$$T_n = \begin{pmatrix} a_1 & c_1 & 0 & \cdots & 0 \\ b_1 & a_2 & c_2 & \ddots & \vdots \\ 0 & b_2 & a_3 & \ddots & \ddots \\ & \ddots & \ddots & \ddots & c_{n-2} & 0 \\ \vdots & & \ddots & b_{n-2} & a_{n-1} & c_{n-1} \\ 0 & \cdots & & 0 & b_{n-1} & a_n \end{pmatrix}. \quad (4.9)$$

Pokud využijeme rozvoje determinantu matice  $T - \lambda I$  podle sloupce a jako první člen vezmeme prvek  $a_n$  získáváme vztah

$$p_n(\lambda) = (a_n - \lambda)p_{n-1}(\lambda) - c_{n-1}b_{n-1}p_{n-2}(\lambda). \quad (4.10)$$

Získáváme tak rekurzivní předpis pro získání charakteristického polynomu třídiagonální matice, pro jehož úplnost je třeba ještě doplnit, že

$$p_0(\lambda) = 1, \quad p_1(\lambda) = a_1 - \lambda. \quad (4.11)$$

Tento postup se může u těchto matic hodit také proto, že pomocí Lanczosovy metody, o které bude řeč později, jsme schopni jiné matice na tento tvar převést při zachování spektra.

**Příklad 4.2** *Nalezněte charakteristický polynom matice*

$$A = \begin{pmatrix} 1 & 3 & 0 & 0 \\ 0 & -2 & 2 & 0 \\ 0 & 1 & -3 & 2 \\ 0 & 0 & 1 & -2 \end{pmatrix}. \quad (4.12)$$

**Řešení:** Podle předpisu (4.10) postupně získáváme polynomy.

$$p_0(\lambda) = 1 \quad (4.13)$$

$$p_1(\lambda) = 1 - \lambda \quad (4.14)$$

$$p_2(\lambda) = (-2 - \lambda)(1 - \lambda) \quad (4.15)$$

$$p_3(\lambda) = (-3 - \lambda)(-2 - \lambda)(1 - \lambda) - 2(1 - \lambda) \quad (4.16)$$

$$p_4(\lambda) = (-2 - \lambda)(-3 - \lambda)(-2 - \lambda)(1 - \lambda) - 2(-2 - \lambda)(1 - \lambda) \quad (4.17)$$

$$= (\lambda^2 + 5\lambda + 6)(\lambda^2 + \lambda - 2) - 2(\lambda^2 + \lambda - 2) \quad (4.18)$$

$$= \underline{\lambda^4 + 6\lambda^3 + 7\lambda^2 - 6\lambda - 8} \quad (4.19)$$

Řešení tohoto polynomu je

$$\lambda_1 = 1, \quad \lambda_2 = -0.4384, \quad \lambda_3 = -4.5616, \quad \lambda_4 = -2. \quad (4.20)$$

## 4.2 Geršgorinova věta

Často se nám může hodit alespoň nějaká informace o spektru, která je na matici okamžitě vidět a pro řešení některých úloh nám může dobře posloužit pouze přibližné určení oblasti, kde se vlastní čísla nacházejí. Tuto informaci nám může poskytnout následující věta.

**Věta 4.2 (Geršgorinova)** Mějme čtvercovou matici  $A$  dimenze  $n$  s komplexními prvky a množinu

$$S = S_1 \cup S_2 \cup \dots \cup S_n, \quad (4.21)$$

kde

$$S_i = \{z \in \mathbb{C}, |z - a_{ii}| \leq \sum_{j=1, j \neq i}^n |a_{ij}|\}, i = 1, \dots, n, \quad (4.22)$$

pak se všechna vlastní čísla matice  $A$  nacházejí v této množině.

**Důkaz.** Mějme vlastní vektor  $v$  matice  $A$  dimenze  $n$ , pak platí  $Av = \lambda v$  a

$$a_{i1}v_1 + a_{i2}v_2 + \dots + a_{ii}v_i + \dots + a_{in}v_n = \lambda v_i \quad (4.23)$$

a  $i$  volíme tak, aby prvek  $v_i$  byl nejvyšší prvek vektoru  $v$  v absolutní hodnotě. Převodem prvku  $a_{ii}v_i$  na pravou stranu získáme rovnici

$$\sum_{j=1, j \neq i}^n a_{ij}v_j = (\lambda - a_{ii})v_i. \quad (4.24)$$

Vpravo máme nyní pouze jeden člen, členy vlevo se však mohou různě sčítat, či odečítat, proto po aplikaci absolutní hodnoty platí

$$\sum_{j=1, j \neq i}^n |a_{ij}||v_j| \geq |\lambda - a_{ii}||v_i|. \quad (4.25)$$

Podělíme celou rovnici  $|v_i|$

$$\sum_{j=1, j \neq i}^n \frac{|a_{ij}||v_j|}{|v_i|} \geq |\lambda - a_{ii}|. \quad (4.26)$$

Jelikož jsme na začátku vybrali  $i$  tak, že člen  $v_i$  je nejvyšší v absolutní hodnotě, víme že

$$\frac{|v_j|}{|v_i|} \leq 1 \quad (4.27)$$

a při odstranění násobení tímto členem platí

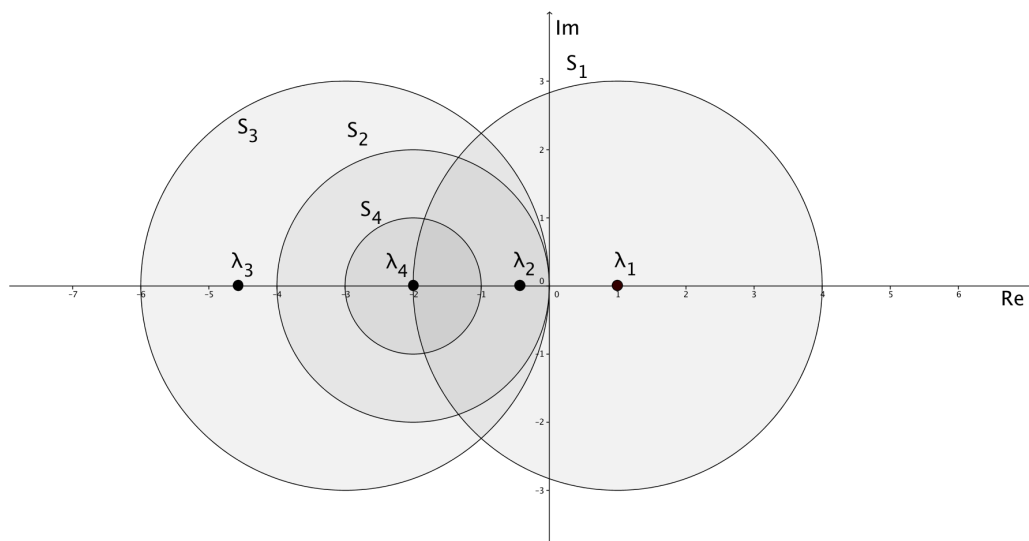
$$\sum_{j=1, j \neq i}^n |a_{ij}| \geq \sum_{j=1, j \neq i}^n \frac{|a_{ij}| |v_j|}{|v_i|} \geq |\lambda - a_{ii}| \quad (4.28)$$

$$\sum_{j=1, j \neq i}^n |a_{ij}| \geq |\lambda - a_{ii}|. \quad (4.29)$$

Pro vlastní číslo  $\lambda$  příslušné vlastnímu vektoru  $v$  tedy platí  $\lambda \in S_i$ . ■

**Příklad 4.3** Pomocí Geršgorinovy věty lokalizujte spektrum matice (4.12).

**Řešení:** Pro každý prvek na hlavní diagonále najdeme množinu  $S_i$  a zakreslíme do obrázku 6. Na obrázku jsou také vyznačena vypočtená vlastní čísla 4.20.



Obrázek 6: Lokalizace spektra

### 4.3 Podobnost matic

U mnoha metod pro výpočet vlastních čísel se využívá podobnost matic. Jsou to například QR metoda popsaná v [3] nebo Lanczosova metoda. Využívá se zde skutečnosti, že podobné matice mají totožné spektrum.

Čtvercové matice  $A$  a  $B$  stejné dimenze  $n$  jsou podobné, pokud platí

$$A = X^{-1}BX, \quad (4.30)$$

kde  $X$  je regulární matice dimenze  $n$ .

**Důkaz.** Mějme dvě podobné matice,  $A$  a  $B$ . Z definice vlastních čísel a pomocí vlastností determinantů můžeme odvodit.

$$0 = \det(A - \lambda I) \quad (4.31)$$

$$= \det(X^{-1}BX - \lambda I) \quad (4.32)$$

$$= \det(X^{-1}BX - X^{-1}\lambda IX) \quad (4.33)$$

$$= \det(X^{-1}(B - \lambda I)X) \quad (4.34)$$

$$= \det(X^{-1}) \det(X) \det(B - \lambda I) \quad (4.35)$$

$$= \det(B - \lambda I) \quad (4.36)$$

Lze tedy vidět, že podobné matice mají stejné spektrum ■

Pomocí této skutečnosti lze jednoduše ukázat, že platí

$$\sigma(AB) = \sigma(BA), \quad (4.37)$$

jelikož matice  $AB$  a  $BA$  jsou podobné.

Za předpokladu, že  $A$  nebo  $B$  je regulární lze napsat

$$AB = ABAA^{-1}, \quad (4.38)$$

případně

$$AB = B^{-1}BAB. \quad (4.39)$$

Lze tedy vidět podobnost matice, kde v (4.38) by  $A^{-1}$  bylo maticí  $X$  v (4.30). V (4.39) je to pak matice  $B$ .

#### 4.4 Mocninná metoda

Mocninná metoda je iterační metoda pro získání vlastního vektoru příslušnému dominantnímu vlastnímu číslu. Dominantním vlastním číslem rozumíme vlastní číslo nejvyšší v absolutní hodnotě. Při studiu technických úloh vedoucích na problém vlastních čísel hrají dominantní vlastní čísla často nejdůležitější roli. Předpokladem mocninné metody je, že matice  $A$  je čtvercová matice řádu  $n$  a pro její vlastní čísla platí

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n| \quad (4.40)$$

a jim příslušné vlastní vektory jsou lineárně nezávislé.

**Algoritmus 4.1** *Algoritmus mocninné metody (Výpis implementace 1)*

1.  $v_1$  = jakýkoliv nenulový počáteční vektor
2. while(chyba > požadovaná přesnost)
  - {
  - $v_i = \frac{1}{\alpha_i} Av_{i-1}$
  - }



kde  $\alpha_i$  je prvek vektoru  $Av_{i-1}$ , který je nejvyšší v absolutní hodnotě. Toto dělení činíme proto, jelikož by se nám při mnohokrát se opakujícím násobení velkých čísel dostávaly hodnoty prvků vlastního vektoru do obrovských čísel a výpočet by se stával nepřesným.

**Důkaz.**<sup>4</sup> Necht' platí (4.40) a  $v_1, v_2, \dots, v_n$  jsou jim příslušné lineárně nezávislé vlastní vektory. Počáteční vektor  $u_0$  volíme náhodně a můžeme jej vyjádřit jako lineární kombinaci vlastních vektorů

$$u_0 = c_1 v_1 + c_2 v_2 + \dots + c_n v_n, \quad (4.41)$$

kde  $c_i$  jsou reálné konstanty, které jsou z důvodu náhodnosti vektoru nenulové s pravděpodobností 1. Lze dále psát

$$A^k u_0 = c_1 A^k v_1 + c_2 A^k v_2 + \dots + c_n A^k v_n \quad (4.42)$$

a z definice vlastních čísel (2.1) víme, že bude platit

$$c_i A^k v_i = c_i \lambda_i^k v_i. \quad (4.43)$$

Můžeme dále upravit rovnici (4.42)

$$A^k u_0 = c_1 \lambda_1^k v_1 + c_2 \lambda_2^k v_2 + \dots + c_n \lambda_n^k v_n \quad (4.44)$$

a po vytknutí  $c_1 \lambda_1^k$  z levé strany dostáváme

$$A^k u_0 = c_1 \lambda_1^k \left( v_1 + \frac{c_2}{c_1} \left( \frac{\lambda_2}{\lambda_1} \right)^k v_2 + \dots + \frac{c_n}{c_1} \left( \frac{\lambda_n}{\lambda_1} \right)^k v_n \right). \quad (4.45)$$

Nyní můžeme jasně vidět nutnost předpokladu (4.40). Jakmile totiž  $k \rightarrow \infty$  a všechny podíly vlastních čísel v závorkách jsou menší než 1, konverguje celá podtržená část k 0, a tudíž za splnění předpokladů mocninné metody lze psát

$$A^k u_0 = c_1 \lambda_1^k v_1. \quad (4.46)$$

Výraz  $A^k u_0$  tudíž konverguje k násobku dominantního vlastního vektoru. ■

#### 4.4.1 Konvergence k dominantnímu vlastnímu číslu

Jak již bylo řečeno, mocninná metoda konverguje k vlastnímu vektoru příslušnému dominantnímu vlastnímu číslu. Pokud by však bylo toto číslo více než jedno, např. 1 a  $-1$ , je jejich absolutní hodnota stejná a metoda nebude konvergovat. Lze proto použít posunutí spektra jedním směrem a získání posunutého vlastního čísla. Posunutí  $B = A + kI$  změní všechna vlastní čísla o  $k$ . Z dominantních vlastních čísel 1 a  $-1$  se při  $k = 1$  stává 2 a 0. Metoda tedy bude opět konvergovat a získáme jí vlastní číslo 2. Nesmíme však od něj zapomenout odečíst 1. Dominantní vlastní čísla se stejnou absolutní hodnotou se nejčastěji vyskytují u matic s komplexními vlastními čísly, jelikož ty se vyskytují v komplexně sdružených párech.

<sup>4</sup>Inspirace na [http://en.wikipedia.org/wiki/Power\\_iteration](http://en.wikipedia.org/wiki/Power_iteration)

#### 4.4.2 Rychlost konvergence

Jak plyne z důkazu mocninné metody, rychlost konvergence je dána poměrem dvou nejvyšších dominantních vlastních čísel.

$$\rho_K = \frac{|\lambda_2|}{|\lambda_1|} \quad (4.47)$$

Při nižší hodnotě této konstanty klesá počet iterací, potřebných k nalezení vlastního čísla. Jako příklad může být matice  $L$  z příkladu 3.1 pro 10 elementů s dominantním vlastním číslem -3,9190. Druhé dominantní vlastní číslo je -3,6825,  $\rho_K$  je tedy asi 0.9397. Mocninná metoda s přesností  $10^{-7}$  konverguje po 230 iteracích. Pokud však posuneme celé spektrum o 1,8 výše, z předchozích vlastních čísel se stane -2.2190 a -1.8825,  $\rho_K = 0.8884$  a metoda se stejnou přesností konverguje po 130 iteracích. Posunutí je tedy dobré odhadnout tak, aby se zachovala dominantní vlastní čísla a zároveň se co nejvíc zmenšila jejich absolutní hodnota.

#### 4.4.3 Získávání dalších vlastních čísel

Mocninná metoda vždy získá dominantní vlastní číslo. Pokud tedy chceme získat další vlastní čísla, stačí původní matici upravit tak, aby se dominantním stalo jiné vlastní číslo.

První způsob je posunutí, které jsme viděli před chvílí. Tímto způsobem však získáme pouze vlastní číslo z druhé strany spektra, jelikož se po posunutí například o největší vlastní číslo toto číslo nuluje, a předchozí nejmenší se stává dominantním.

Druhá možnost je hledání vlastních čísel inverzní matice. Při inverzi proběhne stejná operace i na spektru matice a vlastní číslo, které bylo před inverzí nejbližší nule, se stává dominantním. Kombinací tohoto postupu a posouvání spektra lze získat například vlastní číslo nejbližší číslu 5 a to spočtením vlastního čísla matice  $(A + 5I)^{-1}$ . Postupně lze takto získat všechna vlastní čísla. Postup je však značně neefektivní a nepřesný kvůli inverzi.

Třetí technika je deflace (česky redukce) matice, což je úprava původní matice tak, aby byla zachována všechna vlastní čísla kromě dominantního, které se nuluje. Dominantním se tedy stává předchozí druhé dominantní číslo. Opakováním tohoto postupu lze získat libovolný počet dominantních vlastních čísel.

Jedním z typů deflace symetrických matic je využití ortogonálního projektoru, vytvořeného pomocí získaného vlastního vektoru. Soustředíme se na symetrické matice, jelikož při odvození využijeme ortogonalitu vlastních vektorů. Projektor

$$P = v(v^T v)^{-1} v^T, \quad (4.48)$$

kde jako  $v$  bereme vlastní vektor příslušného vlastního čísla  $\lambda$ , které chceme vynulovat. Je dobré si uvědomit, že  $(v^T v)^{-1}$  je pouze konstanta a výsledkem násobení  $vv^T$  bude matice s násobky onoho vlastního vektoru ve sloupcích. Využijeme spektrální rozklad matice

$$A = QDQ^T, \quad (4.49)$$

kde  $Q$  je matice s ortonormálními vlastními vektory a  $D$  je matice s vlastními čísly na hlavní diagonále. Dále použijeme komplement projektoru  $I - P$  pro získání matice  $B$  se stejnými vlastními čísly, jako  $A$ , až na to dominantní.

$$B = QDQ^T(I - P) \quad (4.50)$$

$$B = QD(Q^T - Q^TP) \quad (4.51)$$

Jelikož se v projektoru nacházejí ve sloupcích násobky vlastního vektoru, ze kterého byl vytvořen, je díky ortogonalitě vlastních vektorů výsledkem násobení  $Q^TP$  matice, která má všechny prvky nulové, pouze na řádku příslušného vlastního vektoru se nachází právě onen normovaný vlastní vektor, který od matice  $Q^T$  odečteme. Tato operace je ekvivalentní vynulování příslušného vlastního čísla v matici  $D$ , neboť právě to by nám vynulovalo příslušný řádek v matici  $Q^T$ . Matice  $B$  má tedy totožné spektrum kromě dominantního vlastního čísla.

Matici  $B$  můžeme dále získat i bez přenásobování a ušetřit tak čas při výpočtu. Lze si uvědomit

$$B = A(I - P) \quad (4.52)$$

$$B = A - AP \quad (4.53)$$

$$B = \underline{A - \lambda P}. \quad (4.54)$$

Využijeme zde opět toho, že v projektoru se nachází ve sloupcích násobky vlastního vektoru a podle definice vlastních vektorů jsou operace  $AP$  a  $\lambda P$  ekvivalentní.

Pomocí předchozích metod můžeme navrhnout následující algoritmus pro výpočet prvních  $n$  dominantních vlastních čísel a vektorů matice.

**Algoritmus 4.2** Výpočet  $n$  dominantních vlastních čísel a vektorů (Výpis implementace 6):

```

1. for(i = 1, ..., n)
    {
        vlastniCislo(i), vlastniVektor(i) = MocninnaMetoda(A)
        A = A - vlastniCislo(i) · P
    }

```

Získáváme tak metodu, která může být velice efektivní při potřebě získat jen několik dominantních vlastních čísel matice. Problém však může být u velkých, řídkých matic. I při řídkých maticích je totiž projektor plná matice a nemusí se nám vejít do paměti. Další problém může být šíření chyb. Nepřesnost při výpočtu prvních vlastních vektorů se skrz projektor nadále šíří a zvětšuje do dalších vlastních vektorů.

#### 4.4.4 Iterace Rayleighova kvocientu

Tato metoda může být zajímavou úpravou mocninné metody za dalšího předpokladu symetrické matice. Metoda využívá inverzních iterací a konverguje k vlastnímu vektoru, který je nejbližší počátečnímu vektoru, vstupujícímu na začátku do algoritmu.

**Algoritmus 4.3** *Algoritmus iterace Rayleighova kvocientu:*

1.  $v_1 =$  počáteční vektor normy 1
2. while(chyba > požadovaná přesnost)
  - {
  - $\sigma_i = (Av_{i-1}, v_{i-1})$
  - $v_i = \frac{1}{\alpha_i} (Av_{i-1} - \sigma_i I)^{-1} v_{i-1}$
  - }

kde  $\alpha_i$  volíme tak, aby byla norma  $v_i$  1.

Metoda konverguje podstatně rychleji, než klasická mocninná metoda, využívá však výpočetně velmi drahou a nepřesnou inverzi matice. Toto omezení lze však odstranit úpravou druhé části algoritmu přenásobením  $(Av_{i-1} - \sigma_i I)$  zleva. Získáváme tak soustavu rovnic

$$(Av_{i-1} - \sigma_i I)v_i = v_{i-1} \quad (4.55)$$

a v každém kroku algoritmu tak nemusíme řešit inverzní matici, nýbrž pouze soustavu rovnic.

Jelikož potřebujeme na počátku aproximaci vlastního vektoru, je možné zkombinovat mocninnou metodu pro nalezení aproximace s nižší přesností a poté využít metody iterace Rayleighova kvocientu. Algoritmus vypadá následovně:

**Algoritmus 4.4** *Kombinace mocninné metody a iterace Rayleighova kvocientu (Výpis implementace 2):*

1.  $v_1 =$  počáteční vektor
2. while(chyba > přesnost aproximace)
  - {
  - $v_i = \text{MocninnaMetoda}(A)$
  - }
3. while(chyba > požadovaná přesnost)
  - {
  - $v_i = \text{RayleighovyIterace}(A)$
  - }

Tuto úpravu mocninné metody můžeme také využít v algoritmu 4.2.

## 4.5 Lanczosova metoda

Získávání vlastních čísel může být velmi efektivní pro speciální typy matic i velmi vysokých dimenzí. Je proto dobré tyto typy znát a ještě lépe umět jiné matice na tyto typy převést při zachování spektra. Lanczosova metoda za pomoci podobnosti matic převádí původní matici na symetrickou, třídiagonální matici a jak jsme ukázali v kapitole 4.3, podobné matice mají stejné spektrum.

#### 4.5.1 Definice

Mějme reálnou, symetrickou, pozitivně definitní matici  $A$ , ortonormální matici  $Q$  a třídiagonální matici  $T$  ve tvaru

$$T = \begin{pmatrix} a_1 & b_1 & 0 & \cdots & 0 \\ b_1 & a_2 & b_2 & \ddots & \vdots \\ 0 & b_2 & a_3 & \ddots & \ddots \\ & \ddots & \ddots & \ddots & b_{n-2} & 0 \\ \vdots & & \ddots & b_{n-2} & a_{n-1} & b_{n-1} \\ 0 & \cdots & & 0 & b_{n-1} & a_n \end{pmatrix}$$

a platí

$$T = Q^T A Q. \quad (4.56)$$

Matice  $A$  a  $T$  jsou potom podobné a mají stejné spektrum.

#### 4.5.2 Odvození

Označme si  $i$ -tý sloupec matice  $Q$  jako  $s_i$  a rovnici (4.56) přenásobme zleva maticí  $Q$ , získáváme vztah

$$(s_1, s_2, \dots, s_n) \begin{pmatrix} a_1 & b_1 & \cdots & 0 \\ b_1 & a_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & b_{n-1} \\ 0 & \cdots & b_{n-1} & a_n \end{pmatrix} = A(s_1, s_2, \dots, s_n). \quad (4.57)$$

Pro  $k$ -tý sloupec matice  $AQ$  platí vztah

$$b_{k-1}s_{k-1} + a_k s_k + b_k s_{k+1} = A s_k \quad (4.58)$$

a pro okrajové sloupce je ještě potřeba dodefinovat

$$b_0 = 0, \quad b_n = 0, \quad s_0 = 0, \quad s_{n+1} = 0. \quad (4.59)$$

V následujících úpravách budeme využívat ortonormality sloupců matice  $Q$ , kde  $(s_i, s_i) = 1$  a  $(s_i, s_j) = 0$  pokud  $i \neq j$ . Přenásobíme-li rovnici (4.58) skalárně vektorem  $s_i$  zprava, získáme vztah pro koeficient  $a_i$ .

$$b_{k-1}(s_{k-1}, s_k) + a_k(s_k, s_k) + b_k(s_{k+1}, s_k) = (A s_k, s_k) \quad (4.60)$$

$$a_k = (A s_k, s_k) \quad (4.61)$$

Dalším přenásobením rovnice (4.58) vektorem  $s_{k+1}$  získáváme vztah pro koeficient  $b_i$ .

$$b_{k-1}(s_{k-1}, s_{k+1}) + a_k(s_k, s_{k+1}) + b_k(s_{k+1}, s_{k+1}) = (A s_k, s_{k+1}) \quad (4.62)$$

$$b_k = (A s_k, s_{k+1}) \quad (4.63)$$

Rovnici (4.58) převedeme na tvar

$$b_k s_{k+1} = As_k - a_k s_k - b_{k-1} s_{k-1} \quad (4.64)$$

$$s_{k+1} = \frac{As_k - a_k s_k - b_{k-1} s_{k-1}}{b_k}. \quad (4.65)$$

Vektor  $k$  je ortonormální, a proto platí

$$b_k = \|As_k - a_k s_k - b_{k-1} s_{k-1}\|. \quad (4.66)$$

Získáváme tak ortonormalizační proces pro získání všech sloupců matice  $Q$ .

### 4.5.3 Algoritmus

**Algoritmus 4.5** *Lanczosova metoda (Výpis implementace 3):*

1.  $s_0 = 0, s_1 = (1, 0, 0, \dots, 0), b_0 = 1, i = 1$
2. while( $i = 1, \dots, \text{dimenze matice} - 1$ )
  - {
  - $a_i = (As_i, s_i)$
  - $T_{i,i} = a_i$
  - $l_i = As_i - a s_i - b s_{i-1}$
  - $b_i = \|l_i\|$
  - $T_{i,i+1} = T_{i+1,i} = b_i$
  - $s_{i+1} = l_i / b_i$
  - }
3.  $i = \text{dimenze matice}$
4.  $a_i = (As_i, s_i)$  //poslední koeficient mimo cyklus, nepotřebujeme již  $b$  a  $s_{i+1}$

V [3] nebo [4] se používá úprava výrazu  $As_k - a_k s_k$  na výraz  $(A - a_k I)s_k$ . Pro algoritmus je však výhodnější, pokud se počítá bez tohoto vytknutí, neboť můžeme použít mezivýsledek  $As_i$  dvakrát. V této implementaci se také nepoužívá  $n$  iterací cyklu s kontrolou indexu pro přiřazení hodnoty  $b_i$ , ale pouze  $n - 1$  iterací se získáním posledního koeficientu  $a$  zvlášť. Zbavujeme se tak jednoho výpočtu  $s_{i+1}$  a hlavně kontroly indexu přes if, který je v Matlabu velmi pomalý.

### 4.5.4 Příklad

Aplikujte Lanczosovu metodu na matici

$$A = \begin{pmatrix} 2 & -1 & 1 \\ -1 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix}. \quad (4.67)$$

Následuje postup algoritmu.

$$\begin{aligned}
 s_1 &= (1, 0, 0)^T \\
 i = 1 \quad a_1 &= 2, \\
 l_1 &= (0, -1, 1)^T \\
 b_1 &= \sqrt{2} \\
 s_2 &= \frac{1}{2}(0, -\sqrt{2}, \sqrt{2})^T \\
 i = 2 \quad a_2 &= \frac{3}{2} \\
 l_2 &= -\frac{1}{4}(0, \sqrt{2}, \sqrt{2})^T \\
 b_2 &= \frac{1}{2} \\
 s_3 &= -\frac{1}{2}(0, \sqrt{2}, \sqrt{2})^T \\
 i = 3 \quad a_3 &= \frac{3}{2}
 \end{aligned}$$

Výsledná matice

$$T = \begin{pmatrix} 2 & \sqrt{2} & 0 \\ \sqrt{2} & 1.5 & 0.5 \\ 0 & 0.5 & 1.5 \end{pmatrix} \quad (4.68)$$

má stejné spektrum jako matice  $A$  a můžeme snadno ověřit, že

$$\sigma(A) = \sigma(T) \approx \{0.1981, 1.5550, 3.2470\}. \quad (4.69)$$

## 4.6 Numerické experimenty

Výše jsme již uvedli několik metod algoritmů pro výpočet vlastních čísel a nyní si některé z nich otestujeme. Jako testovací matice nám poslouží matice  $A$ , vznikající při řešení průhybu čtvercové membrány ve 2D metodou sítí, která je například popsána v [4]. Při vytváření matice membrány pro  $n$  dělení vzniká matice dimenze  $n^2$  ve tvaru

$$A = \begin{matrix} & \begin{matrix} a_{i,1} & \dots & a_{i,n} & \dots & a_{i,n^2} \end{matrix} \\ \begin{matrix} a_{1,j} \\ \vdots \\ a_{n,j} \\ \vdots \\ a_{n^2,j} \end{matrix} & \begin{pmatrix} 4 & -1 & 0 & \dots & 0 & -1 & 0 & \dots & 0 \\ -1 & 4 & \ddots & 0 & \dots & 0 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 4 & -1 & 0 & & \ddots & -1 \\ 0 & & 0 & -1 & 4 & -1 & 0 & & 0 \\ -1 & \ddots & & 0 & -1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & & \ddots & \ddots & \ddots & -1 & 0 \\ \vdots & \ddots & -1 & 0 & \dots & 0 & -1 & 4 & -1 \\ 0 & \dots & 0 & -1 & 0 & \dots & 0 & -1 & 4 \end{pmatrix} \end{matrix} .$$

Jelikož se jedná o pozitivně definitní matici, budou její spektrum tvořit pouze kladná, reálná čísla. Toho můžeme využít, pokud budeme chtít vypočítat nejmenší vlastní čísla a pomocí posunutí spektra o dominantní vlastní číslo se nám původně nejnižší vlastní čísla stávají dominantními.

Budeme počítat 5 nejvyšších a 5 nejnižších vlastních čísel uvedené matice různých rozměrů a otestujeme dva následující algoritmy.

**Algoritmus 4.6** *Mocninná metoda s maticovou redukcí a posunem spektra (Výpis implementace 8):*

```

1.  $B = A$ 
2. for( $i = 1, \dots, 5$ )
   {
        $vlastniCislo(i) = MocninnaMetoda(A)$ 
        $A = A - vlastniCislo(i) \cdot P$ 
   }
3.  $B = A - vlastniCislo(1) \cdot I$  //posun spektra o dominantní vlastní číslo
4. for( $i = 1, \dots, 5$ )
   {
        $vlastniCislo(i) = MocninnaMetoda(B) + vlastniCislo(1)$ 
        $A = A - vlastniCislo(i) \cdot P$ 
   }
```

**Algoritmus 4.7** *Iterace Rayleighova kvocientu s předchozí aproximací mocninnou metodou, maticovou redukcí a posunem spektra (Výpis implementace 9):*

```

1.  $B = A$ 
2. for( $i = 1, \dots, 5$ )
   {
        $aproximace = MocninnaMetoda(A, \text{přesnost aproximace})$ 
        $vlastniCislo(i) = RayleighovaIterace(A, aproximace)$ 
        $A = A - vlastniCislo(i) \cdot P$ 
   }
3.  $B = A - vlastniCislo(1) \cdot I$  //posun spektra o dominantní vlastní číslo
4. for( $i = 1, \dots, 5$ )
   {
        $aproximace = MocninnaMetoda(B, \text{přesnost aproximace})$ 
        $vlastniCislo(i) = RayleighovaIterace(B, aproximace)$ 
        $A = A - vlastniCislo(i) \cdot P$ 
   }
```

Přesnost vypočteného vlastního čísla jsme kontrolovali pomocí normy reziduí dvou posledních aproximací vlastního vektoru, přičemž ukončovací rozdíl činil  $10^{-6}$ . V algoritmu 4.7 byla brána ukončovací norma pro aproximaci  $10^{-3}$ . Testovali jsme matice pro 15, 30 a 50 dělení a tedy dimenze 225, 900 a 2500. Očekáváme, že modifikace pomocí Rayleighova



kvocientu v algoritmu 4.7 přinese snížení počtu iterací, nicméně se během něj vytváří matice velmi blízké singulárním a algoritmus je tak třeba ukončovat dříve, což může vést k numerickým nestabilitám. Měřili jsme rychlost a chybu výpočtu, která byla porovnávána oproti standardní funkci Matlabu `eig`. V následujících tabulkách jsou zobrazeny také počty iterací.

	Největší vl. čísla					Nejmenší vl. čísla				
Počet iterací	863	880	838	1260	800	636	1481	747	924	588
Chyba( $\cdot 10^{-5}$ )	0.2231	0.3502	0.1217	0.0372	0.2079	0.2386	0.0792	0.1523	0.2309	0.1295
Čas[s]	0.3735									

Tabulka 1: Výsledky testu algoritmu 4.6 pro matici  $225 \times 225$

	Největší vl. čísla					Nejmenší vl. čísla				
Počet iterací	295	288	512	520	227	146	701	338	366	147
Chyba( $\cdot 10^{-5}$ )	0.000	0.0005	0.2216	0.0000	0.0000	0.0000	0.0000	0.0172	0.0101	0.0025
Čas[s]	0.2003									

Tabulka 2: Výsledky testu algoritmu 4.7 pro matici  $225 \times 225$

	Největší vl. čísla					Nejmenší vl. čísla				
Počet iterací	3307	2972	2464	4472	2877	2130	5505	2964	2400	2059
Chyba( $\cdot 10^{-5}$ )	0.0627	0.6141	0.0609	0.0444	0.0145	0.1000	0.0444	0.2396	0.0401	0.1417
Čas[s]	5.694									

Tabulka 3: Výsledky testu algoritmu 4.6 pro matici  $900 \times 900$

	Největší vl. čísla					Nejmenší vl. čísla				
Počet iterací	1186	1093	745	1603	1235	330	2510	1387	1179	467
Chyba( $\cdot 10^{-7}$ )	0.000	0.0000	0.0000	0.0000	0.0000	0.0264	0.0000	0.0000	0.0000	0.3308
Čas[s]	3.6099									

Tabulka 4: Výsledky testu algoritmu 4.7 pro matici  $900 \times 900$

	Největší vl. čísla					Nejmenší vl. čísla				
Počet iterací	9527	7554	8905	11751	8177	5032	15012	6231	7664	4730
Chyba( $\cdot 10^{-5}$ )	0.0291	0.1922	0.0379	0.0272	0.1570	0.0870	0.0272	0.3948	0.0353	0.0534
Čas[s]	225.7906									

Tabulka 5: Výsledky testu algoritmu 4.6 pro matici  $2500 \times 2500$

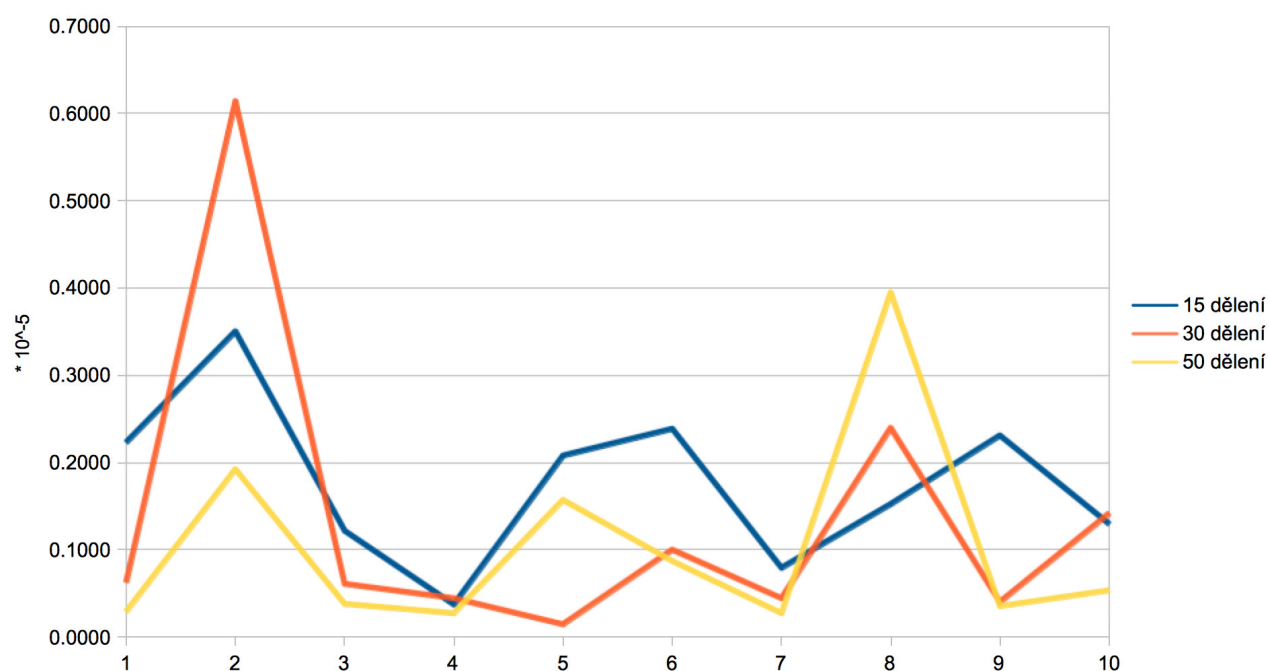
	Největší vl. čísla					Nejmenší vl. čísla				
Počet iterací	2447	1915	2123	3178	3136	819	7021	3468	3007	1145
Chyba ( $\cdot 10^{-7}$ )	0.000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.2245
Čas [s]	84.0003									

Tabulka 6: Výsledky testu algoritmu 4.7 pro matici  $2500 \times 2500$

Z vypočtených hodnot vidíme, že modifikace pomocí iterace Rayleighova kvocientu urychluje výpočet, neboť dosažení aproximace mocninnou metodou potřebuje méně iterací a metoda iterace Rayleighova kvocientu poté konverguje velmi rychle. Problémem však je, že při výpočtu generuje matice velice blízké singulárním a ve výpočtu se již z tohoto důvodu nepokračuje. Můžeme to vidět na chybách, kde se u této metody vyskytuje jedna nebo více „dominujících“ chyb, vůči kterým jsou jiné chyby zanedbatelně malé.

Projevila se také nevýhoda redukce spektra matice pomocí projektoru. Pokud máme původní matici jako řídkou, probíhá získání dominantního vlastního čísla mnohonásobně rychleji díky neustálému násobení, které je pro řídké matice efektivnější. Redukcí spektra pomocí projektoru však tuto řídkost ztratíme a musíme dále počítat s plnou maticí. Pro využití této metody pro počítání více vlastních čísel velkých řídkých matic by tedy bylo potřeba navrhnout jiný způsob redukce, zachovávající řídkost.

Na grafu níže si můžeme prohlédnout chyby pro algoritmus 4.6.



Obrázek 7: Chyby výpočtu algoritmu 4.6

#### 4.6.1 Modifikace pro řídké matice

Jak jsme viděli, použití projektoru s sebou nese nevýhodu ztráty řídkosti pro řídké matice a pro matice vysokých dimenzí se již stává tato metoda nepoužitelnou. Prozatím jsme počítali pouze s úpravou matice  $A$  a její modifikací podle rovnice (4.54) a zde nám také vzniká neřídkost. Je však možné mocninnou metodu upravit tak, aby stále počítala s řídkou maticí  $A$  a informaci o projektoru ji navíc dodat tak, aby se projektor při výpočtech projevil, nicméně nebylo nutné ho celý sestavovat a počítat tak s velkou, neřídkou maticí.

Předpokládejme, že již máme spočítanou dominantní vlastní dvojici  $(\lambda_1, v_1)$  a pokud tím pádem aplikujeme mocninnou metodu na matici

$$B = A - \lambda_1 v_1 (v_1^T v_1) v_1^T, \quad (4.70)$$

získáme druhé dominantní vlastní číslo. Můžeme však zkusit upravit základní předpis mocninné metody

$$u_{i+1} = B u_i \quad (4.71)$$

$$u_{i+1} = (A - \lambda_1 v_1 (v_1^T v_1) v_1^T) u_i \quad (4.72)$$

$$u_{i+1} = \underline{A u_i - \lambda_1 v_1 (v_1^T v_1) (v_1^T u_i)}. \quad (4.73)$$

Výsledkem násobení  $v_1^T v_1$  a  $v_1^T u_i$  jsou skaláry a v obou případech násobíme pouze dva vektory a poté konstantu a vektor. Vyhneme se tak ztrátě řídkosti matice a tato operace navíc je v porovnání s násobením matic zanedbatelná. Pokud bychom chtěli spočítat třetí dominantní vlastní číslo, přibude do předpisu (4.73) pouze výraz  $-\lambda_2 v_2 (v_2^T v_2) (v_2^T u_i)$  a obdobně bychom pokračovali pro další vlastní čísla.

**Algoritmus 4.8** Mocninná metoda s projekcí (Výpis implementace 10 pro  $n$  vlastních čísel)

1.  $u_1$  = jakýkoliv nenulový počáteční vektor
2.  $\lambda_1, \dots, \lambda_k, v_1, \dots, v_k$  = předchozí dominantní vlastní čísla a vektory
3. while(chyba > požadovaná přesnost)
 

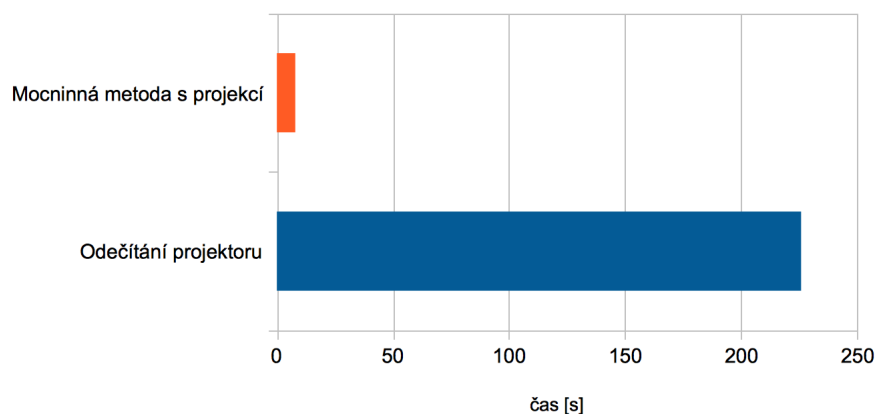
$$\left\{ \begin{array}{l} u_i = \frac{1}{\alpha_i} A u_{i-1} - \sum_{j=1}^k \lambda_j v_j (v_j^T v_j) (v_j^T u_{i-1}) \end{array} \right.$$

Otestovali jsme tuto metodu pro získání 5 nejvyšších a 5 nejnižších vlastních čísel na stejné matici a stejných podmínkách, jako předchozí testy pro matici  $2500 \times 2500$  a získali jsme výsledky, zapsané v tabulce 7.

	Největší vl. čísla					Nejmenší vl. čísla				
Počet iterací	8128	7589	8475	11660	7389	4598	15738	8246	6620	4759
Chyba ( $\cdot 10^{-5}$ )	0.0825	0.0289	0.1611	0.0372	0.0533	0.1023	0.0436	0.3880	0.0312	0.0042
Čas[s]	7.9481									

Tabulka 7: Výsledky testu algoritmu 4.8 pro matici  $2500 \times 2500$

Všimněme si obrovského rozdílu v čase výpočtu oproti tabulce 6, zobrazeného na obrázku 8. Při zvyšující se dimenzi matice je tento rozdíl stále výraznější díky složitosti násobení plné matice ( $n^3$ ). Podařilo se nám zamezit ztrátě řádkosti matice a díky této optimalizaci je získání dominantních vlastních čísel tímto algoritmem velmi efektivní pro velké, řídké matice.



Obrázek 8: Porovnání časů algoritmu 4.8 a 4.6

## 5 Další výzkum - Předpodmíněné řešiče vlastních čísel

Následující kapitola slouží jako nástin problematiky, která bude předmětem výzkumu v navazujícím studiu.

### 5.1 Předpodmínění soustav lineárních rovnic

Předpodmínění je dlouho známou technikou vyvinutou zejména pro řešení velkých soustav lineárních rovnic, jehož účelem je zvýšit rychlost konvergence gradientní iterační metody k řešení soustavy. Princip předpodmínění spočívá v převodu původní soustavy  $Ax = b$  pomocí přenásobení maticí  $B^{-1}$  na soustavu se stejným řešením avšak s maticí mající nižší číslo podmíněnosti, ideálně blízkí se 1. Jako matice  $B$  se volí taková matice, která je blízká  $A$  a zároveň jsme schopni za rozumnou dobu nalézt inverzi. Poté řešíme následující soustavu

$$B^{-1}Ax = B^{-1}b \quad (5.1)$$

$$\bar{A}x = \bar{b} \quad (5.2)$$

O čísle podmíněnosti a předpodmínění soustav lineárních rovnic si lze přečíst například v [3].

### 5.2 Předpodmínění u vlastních čísel

Využití předpodmínění při řešení problému vlastních čísel je však poměrně novou věcí a není ještě podrobně prozkoumáno. Využívá se zde poznatků získaných při vývoji metod pro řešení soustav lineárních rovnic, které jsou aplikovány na problém vlastních čísel. Mnoho informací shromáždil Andrew V. Knyazew v publikaci [6], kde dává dohromady mnoho nalezených poznatků zejména ze sovětské literatury a přidává k nim své vlastní, získané při vývoji předpodmíněných řešičů. Pomocí nejnovějších metod bylo ukázáno, že vlastní dvojici lze získat za přibližně stejný počet operací, jako vyřešení soustavy lineárních rovnic stejných rozměrů.

Základní myšlenka zní, že abychom zrychlili konvergenci daných metod, je třeba předpodmíněním změnit spektrum pro zajištění rychlejší konvergence metod. Jako metody s předpodmíněním se využívají upravená iterace Rayleighova kvocientu, metoda sdružených gradientů, inverzní iterace mocninné metody, které jsou popsány v [7]. Často používaná je také Davidsonova metoda, která je podrobně popsána a otestována s různými druhy předpodmínění v [8]. V [6] je porovnání metody nejvyššího spádu a metody sdružených gradientů, která v tomto porovnání vítězí a v současné době je jedna z nejpopulárnějších metod pro předpodmíněování problémů vlastních čísel a je dále zkoumána. Téma výpočtu vlastních čísel pomocí předpodmínění je stále ještě z velké části neprobádanou oblastí, o čemž svědčí také fakt, že v češtině se mi k této problematice nepodařilo najít prakticky žádné informace a je zde tedy velký prostor pro výzkum, neboť se zdá, že v předpodmínění řešičů vlastních čísel je velký potenciál.

## 6 Závěr

Cílem práce bylo seznámení se s problémem vlastních čísel, jeho aplikace na reálné problémy, představení některých metod a úvod do problematiky předpokládaných řešičů tohoto problému.

V první části jsme uvedli tři příklady využití vlastních čísel a jejich dopad na daný problém. Tyto ukázky jsou pouze jedny z mnoha a demonstrují, že vlastní čísla nejsou pouze teoretický problém a motivací jejich výzkumu jsou zejména jejich praktické aplikace v mnoha oborech.

V druhé části jsme představili některé metody řešící problém vlastních čísel, provedli jejich implementaci a některé z nich otestovali. Při testech se projevilo, jak důležité jsou vlastnosti matic, které je třeba spolu s problémem vlastních čísel studovat, a to symetričnost, pozitivní definitnost nebo řidkost matice, které výrazně ovlivňují rychlost výpočtu a také to, zda lze vůbec daný algoritmus na metody aplikovat.

V poslední části jsme nastínili možnost předpokládání některých řešičů vlastních čísel, které jsou současným předmětem výzkumu mnoha matematických ústavů a jejichž výzkum může být předmětem diplomové práce navazujícího studia.

Práce může být hodnocena jako seznámení se s problémem vlastních čísel a ukazuje, o jak širokou a zajímavou tematiku se jedná. Z tohoto důvodu je třeba problému vlastních čísel nadále věnovat pozornost, neboť se díky tomu můžeme dozvědět mnoho věcí o systémech a dějích, které jsou kolem nás.

## 7 Reference

- [1] SAAD, Y. *Numerical methods for large eigenvalue problems*. Rev. ed. Philadelphia: Society for Industrial and Applied Mathematics, c2011, xvi, 276 p. ISBN 978-161-1970-722.  
Dostupné z: [http://www-users.cs.umn.edu/~saad/eig\\_book\\_2ndEd.pdf](http://www-users.cs.umn.edu/~saad/eig_book_2ndEd.pdf)
- [2] REKTORYS, K. ČVUT. *Matematika V. Obyčejné parciální a diferenciální rovnice*. Praha: ČVUT, 1989.
- [3] VONDRÁK, V., POSPÍŠIL L. VŠB - TUO. *Numerické metody 1*. Ostrava: Matematika pro inženýry 21. století, 2011.  
Dostupné z: [http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/numericke\\_metody.pdf](http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/numericke_metody.pdf)
- [4] KOZUBEK, T., BRZOBOHATÝ, T., HAPLA, V., JAROŠOVÁ, M., MARKOPOULOS, A. VŠB - TUO. *Lineární algebra s Matlabem*. Ostrava: Matematika pro inženýry 21. století, 2012.  
Dostupné z: [http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/linearni\\_algebra\\_s\\_matlabem.pdf](http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/linearni_algebra_s_matlabem.pdf)
- [5] DOSTÁL, Z., VONDRÁK, V. VŠB - TUO. *Lineární algebra*. Ostrava: Matematika pro inženýry 21. století, 2012.  
Dostupné z: [http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/linearni\\_algebra.pdf](http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/linearni_algebra.pdf)
- [6] KNYAZEV, A. *Preconditioned Eigensolvers - An Oxymoron?*. DOI: 10.1.1.34.9249.  
Dostupné z: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.34.9249>
- [7] KRESSNER, D. *Numerical solution of matrix eigenvalue problems Part 3: Preconditioned Eigensolvers*. Zürich, 2008.  
Dostupné z: <http://www.math.ethz.ch/~kressner/zss/lecture3.pdf>
- [8] STATHOPOULOS, A., SAAD, Y., FISCHER, CH. *Robust Preconditioning of Large Sparse, Symmetric Eigenvalue Problems*. 1994.  
Dostupné z: <http://www-users.cs.umn.edu/~saad/PDF/ahpcrc-93-093.pdf>
- [9] SO, P. *course materials for 2.003J/1.053J Dynamics and Control I*, Spring 2007. MIT OpenCourseWare (<http://ocw.mit.edu>), Massachusetts Institute of Technology.

## A Zdrojové kódy v Matlabu

Následují výpisy naprogramovaných metod pro výpočet vlastních čísel. Kódy lze nalézt i na přiloženém CD.

### A.1 Základní implementace vybraných metod

---

```

function [ eigenvector, iterations ] = GetHighestEigPowerMethod( A, eps )
%GetHighestEigPowerMethod Counts dominating eigenvector with specified
%precision. Returns vector and number of iterations to get it .

dim = size(A,1); % dimension of matrix
vEig = rand(dim, 1); % initial vector
vPrevious = 0*vEig; %next vector where value from previuos iteration is held, zero vector at start

maxIterations = 1000000; % to prevent infinite loop
iterations = 0; % iterations counter

%until the specified precision is reached
while(norm(vEig-vPrevious) > eps && iterations < maxIterations)
    vPrevious = vEig;
    vx = A*vEig; %power method algorithm

    %get the divisor
    max1 = max(vx);
    maxabs = max(abs(vx));
    if (max1 - maxabs < 0)
        alfa = (-1)* maxabs; %max absolute was negative
    else
        alfa = max1;
    end

    vEig = vx/ alfa ; %power method algorithm
    iterations = iterations + 1;
end
%found eigenvector will be returned
eigenvector = vEig;
end

```

---

Výpis 1: Algoritmus mocninné metody



---

```

function [ eigenvector, iterations , raylightIterations ] = GetHighestEigRayleigh( A, eps,
    raylightEps )
%GETHIGHESTEIGRAYLIGHT Counts dominating eigenvector with Raylight quotient with specified
%precision. Returns vector and number of iterations to get it .

dim = size(A,1); % dimension of matrix
maxIterations = 100000; % to prevent infinite loop
raylightIterations = 0; % raylight iterations counter
I = eye(dim); %identity matrix

[vEig, iterations ] = GetHighestEigPowerMethod(A, raylightEps); %get power method approximation
vPrevious = 0*vEig; %next vector where value from previous iteration is held, zero vector at start
vEig = vEig/norm(vEig); % must be size of 1
norme = norm(vEig-vPrevious); %initialization of norm

%until the specified precision is reached
while(norme > eps && iterations < maxIterations)
    iterations = iterations + 1;
    raylightIterations = raylightIterations + 1;
    vPrevious = vEig;
    raylightQuotient = (A*vPrevious)'*vPrevious;
    Aray = (A-raylightQuotient*I);

    % if the raylight quotient make matrix singular so it is an eigenvalue
    if (rcond(Aray) < 1e-15)
        eigenvector = vEig;
        fprintf ('RayleighMethod_-_matrix_close_to_singular\n')
        return;
    end

    vEig = Aray\vPrevious;
    vEig = vEig/norm(vEig); %size must be one again
    norme = norm(vEig-vPrevious);
end
eigenvector = vEig;
end

```

---

Výpis 2: Algoritmus Iterace Rayleighova kvocientu s aproximací mocninnou metodou

---

```

function [ T, Q ] = Lanczos( A )
%LANCZOS Perform Lanczos algorithm on input matrix
% Transform symetric positive definite square matrix to symmetric
% three-diagonal matrix.

dim = size(A,1);

%allocation of T and Q
T(dim, dim) = 0;
Q(dim, dim) = 0;

s = zeros(dim, 1); %s_1 will be unit vector with one on s_1,1
s_minus1 = s; %is used to store previous vector
s(1,1) = 1;
b = 1; %initial b is set to one
Q(1:dim, 1) = s;

for i = 1 : (dim - 1);
    As_i = A*s;
    a = As_i' * s; %(As_i, s_i)
    T(i, i) = a;
    l = As_i - a*s - b*s_minus1;
    s_minus1 = s;
    b = norm(l);
    T(i+1, i) = b;
    T(i, i+1) = b;
    s = l/b;
    Q(1:dim, i + 1) = s;

end

%last step
i = dim;
As_i = A*s;
a = As_i' * s; %(As_i, s_i)
T(i, i) = a;
end

```

---

Výpis 3: Lanczosova metoda

## A.2 Získání více dominantních vlastních čísel

### A.2.1 Pomocné metody

---

```

function [ eig ] = GetEigFromMatrixAndVector( A, eigenvector )
%simply multiply the matrix with eigenvector and we have the eigenvalue
Aeig = A*eigenvector;
eig = Aeig(1)/eigenvector(1);
end

```

---

Výpis 4: Získání vlastního čísla z matice a vektoru

---

```

function [ P ] = GetProjector( v )
%Projector definition
P = v*inv(v'*v)*v';
end

```

---

#### Výpis 5: Výpočet projektoru

### A.2.2 Mocninná metoda

---

```

function [ eigs, iterations , eigenvectors ] = PowerMethodSolve( A, n, eps)
%POWERMETHODSOLVE Counts n dominating eigenvalues with eps precision

dim = size(A,1); %dimension of matrix
eigs(n) = 0; % allocation
eigenvectors(dim,n) = 0; % allocation
iterations (n) = 0; % allocation

% count first eigenvector and corresponding eigenvalue
[eigenvectors(1:dim, 1), iterations (1)] = GetHighestEigPowerMethod(A, eps);
eigs(1) = GetEigFromMatrixAndVector(A, eigenvectors(1:dim,1));
fprintf( ' Power_method_ _Next_eigenpair_found.\n');
Ai = A;

%then use projector to find other eigenvalues
for i = 2 : n
    P = GetProjector(eigenvectors(1:dim, i - 1));
    Ai = Ai - (eigs(i-1) * P);
    [eigenvectors(1:dim, i) , iterations ( i )] = GetHighestEigPowerMethod(Ai, eps);
    eigs(i) = GetEigFromMatrixAndVector(Ai, eigenvectors(1:dim,i));
    fprintf( ' Power_method_ _Next_eigenpair_found.\n');
end
end

```

---

#### Výpis 6: Získání $n$ dominantních vlastních čísel a vektorů mocninnou metodou

### A.2.3 Iterace Rayleighova kvocientu

---

```

function [ eigs, iterations , raylightIterations , eigenvectors ] = RayleighMethodSolve( A, n, eps,
    raylightEps)
%RAYLIGHTMETHODSOLVE Counts n dominating eigenvalues with eps precision

dim = size(A,1); %dimension of matrix
eigs(n) = 0; % allocation
eigenvectors(dim,n) = 0; % allocation
iterations (n) = 0; % allocation
raylightIterations (n) = 0; % allocation

% count first eigenvector and corresponding eigenvalue
[eigenvectors(1:dim, 1), iterations (1), raylightIterations (1)] = GetHighestEigRayleigh(A, eps,
    raylightEps);
eigs(1) = GetEigFromMatrixAndVector(A, eigenvectors(1:dim,1));
fprintf ( ' Rayleigh_method_-_Next_eigenpair_found.\n');
Ai = A;

%then use projector to find other eigenvalues
for i = 2 : n
    P = GetProjector(eigenvectors(1:dim, i - 1));
    Ai = Ai - (eigs(i-1) * P);
    [eigenvectors(1:dim, i), iterations (i), raylightIterations (i)] = GetHighestEigRayleigh(Ai, eps,
        raylightEps);
    eigs(i) = GetEigFromMatrixAndVector(Ai, eigenvectors(1:dim,i));
    fprintf ( ' Rayleigh_method_-_Next_eigenpair_found.\n');
end
end

```

---

Výpis 7: Získání  $n$  dominantních vlastních čísel a vektorů iterací Rayleighova kvocientu

### A.2.4 Největší a nejmenší vlastní čísla

---

```

function [ eigs, iterations ] = PowerMethodSolveBigSmall( A, n, eps )
%POWERMETHODSOLVEBIGSMALL Counts n smallest and n biggest eigenvalues
% This function uses power method to count n dominating eigenvalue of positive definite matrix
% and
% then power method with shift to count n smallest eigenvectors

dim = size(A,1); %dimension of matrix

%allocation
eigs(2*n, 1) = 0;
iterations(2*n, 1) = 0;

[e, it] = PowerMethodSolve(A, n, eps);
eigs(1:n, 1) = e;
iterations(1:n, 1) = it ;

%matrix is positive definite – if we shift that by dominating eigenvalue, teh smallest
%eigenvalues become dominant
dominEig = eigs(1,1);

A_shifted = A – dominEig * speye(dim);

% Use again power method
[e, it] = PowerMethodSolve(A_shifted, n, eps);

% Remove eigenvalue shift
unshift = dominEig;

%Added in reverse order to keep eigenvalues sorted descendend
for i = 1:n
eigs(n + i) = e(n + 1 – i) + unshift;
iterations(n + i) = it(n + 1 – i);
end
end

```

---

Výpis 8: Získání  $n$  největších a nejmenších vlastních čísel a vektorů mocninnou metodou a posunutím

---

```

function [ eigs, iterations ] = RayleighMethodSolveBigSmall( A, n, eps, raylightEps )
%RAYLEIGHMETHODSOLVEBIGSMALL Counts n smallest and n biggest eigenvalues
% This function uses power method to count n dominating eigenvalue of positive definite matrix
% and
% then power method with shift to count n smallest eigenvectors
dim = size(A,1); %dimension of matrix

%allocation
eigs(2*n, 1) = 0;
iterations(2*n, 1) = 0;

[e, it] = RayleighMethodSolve(A, n, eps, raylightEps);
eigs(1:n, 1) = e;
iterations(1:n, 1) = it ;

%matrix is positive definite – if we shift that by dominating eigenvalue, teh smallest
eigenvalues become dominant
dominEig = eigs(1,1);
A_shifted = A – dominEig * eye(dim);

% Use again power method
[e, it] = RayleighMethodSolve(A_shifted, n, eps, raylightEps);

% Remove eigenvalue shift
unshift = dominEig;

%Added in reverse order to keep eigenvalues sorted descendend
for i = 1:n
eigs(n + i) = e(n + 1 – i) + unshift;
iterations(n + i) = it(n + 1 – i);
end
end

```

---

Výpis 9: Získání  $n$  největších a nejmenších vlastních čísel a vektorů Iterací Rayleighova kvocientu a posunutím

---

```

function [ eigs, alliterations , eigenvectors ] = PowerMethodSolveSparse( A, n, eps )
%POWERMETHODSOLVESPARSE Counts n dominating eigenvalues with eps precision

A = sparse(A); %if its already sparse, this do nothing
dim = size(A,1); %dimension of matrix
eigs(n) = 0; % allocation
eigenvectors(dim,n) = 0; % allocation
alliterations (n) = 0; % allocation

% count first eigenvector and corresponding eigenvalue
[eigenvectors(1:dim, 1), alliterations (1)] = GetHighestEigPowerMethod(A, eps);
eigs(1) = GetEigFromMatrixAndVector(A, eigenvectors(1:dim,1));
fprintf ( ' Power_method_sparse_ _Next_eigenpair_found.\n');
maxIterations = 1000000; % to prevent infinite loop

%changed power method algorithm
for i = 2:n
    v = eigenvectors(:, i-1); %previous eigenvector
    e = eigs(i-1); %previous eigenvalue
    con(:, i-1) = (e/(v'*v))*v; %projector and lamda constant modification
    tempEigVecs(i-1, :) = v'; %transposed eigenvectors to store and reach max speed in projector
    apply
    iterations = 0; % iterations counter
    vEig = rand(dim, 1); % initial vector
    vPrevious = 0*vEig; %next vector where value from previuos iteration is held, zero vector at
    start

    while(norm(vEig-vPrevious) > eps && iterations < maxIterations) %until the specified precision
    is reached
        vPrevious = vEig;
        vx = A*vEig - con*(tempEigVecs*vEig); %power method with projector algorithm

        %get the divisor
        max1 = max(vx);
        maxabs = max(abs(vx));
        if (max1 - maxabs < 0)
            alfa = (-1)* maxabs; %max absolute was negative
        else
            alfa = max1;
        end

        vEig = vx/ alfa ; %power method algorithm
        iterations = iterations + 1;
    end

    eigs(i) = GetEigFromMatrixAndVector(A, vEig);
    eigenvectors(:, i) = vEig;
    alliterations (i) = iterations ;
    fprintf ( ' Power_method_sparse_ _Next_eigenpair_found.\n');
end
end

```

---

Výpis 10: Získání  $n$  největších vlastních čísel mocninnou metodou pro řídké matice